



REST API

Sumário

Introdução	5
Sobre a nova versão da API	5
Definições	5
<i>Zuri Settings</i>	6
<i>Theme</i>	7
<i>Menu</i>	8
<i>Ponto no Mapa</i>	8
<i>Processo</i>	9
<i>Link</i>	10
<i>Item na caixa de entrada, meus itens ou acompanhamento</i>	11
<i>SLA</i>	12
<i>GPS</i>	12
<i>GPS Resource</i>	13
<i>Ítem</i>	13
<i>Dados de Execução</i>	15
<i>Tomada de Decisão</i>	16
<i>Decisão</i>	16
<i>Definição de formulário</i>	17
<i>Campo de Formulário</i>	17
Campo Action	18
Campo Boolean	19
Campo Date	19
Campo DateTime	19
Campo Time	19
Campo DateTimeTime	19
Campo Label	20
Campo Position	20
Campo Map	20
Campo Scale	21
Campo Rating	21

Campo Option _____	21
Campo Multiple _____	22
Campo Text _____	22
Campo Group _____	23
<i>Item de campo de opções</i>	<i>24</i>
<i>Sources de formulário</i>	<i>24</i>
<i>Dados de formulário</i>	<i>25</i>
<i>Valores de Campo</i>	<i>26</i>
Campo Action _____	26
Campo Text _____	26
Campo Int _____	26
Campo Decimal _____	26
Campo Boolean _____	26
Campo Scale _____	26
Campo Rating _____	26
Campo Date _____	26
Campo Datetime _____	27
Campo Datetimetime _____	27
Campo Option _____	27
Campo Multiple _____	27
Campo Position _____	27
<i>Item Detalhado</i>	<i>27</i>
<i>Item Histórico</i>	<i>32</i>
<i>Formulário Processo</i>	<i>32</i>
Exemplo do campo zuriKey _____	33
Opções de autenticação _____	33
Identificação do Processo _____	35
<i>Campo de formulário</i>	<i>35</i>
Formulário preenchido _____	36
Formulário preenchido com campos filhos _____	37
<i>Item Processo Formulário</i>	<i>39</i>
Exemplos de respostas _____	40

Tabelas de Tipos --- 40

<i>HTTP Status Code</i>	40
<i>Tipos de Menu</i>	41
<i>Tipos de objeto Theme</i>	41
<i>Tipos de Execução</i>	42
<i>Estado de Execução</i>	42
<i>Tipo de Campo</i>	42

API --- 43

<i>Identificação do device</i>	43
<i>Autenticação</i>	44
<i>Configurações Mobile</i>	45
<i>Pontos no Mapa</i>	45
<i>Lista de Processos</i>	46
<i>Lista de itens na Caixa de Entrada</i>	46
<i>Lista de itens em “Meus itens”</i>	46
<i>Lista de itens em Acompanhamento</i>	47
<i>Detalhe de item na caixa de entrada</i>	47
<i>Executar Ítem</i>	48
<i>Iniciar Processo</i>	48
<i>Listar formulários que podem ser preenchidos offline</i>	48
<i>Atualizar dados de formulário já preenchido (checkout)</i>	48
<i>Atualização de sources de formulário</i>	49
<i>Detalhes de histórico de um item de processo</i>	49
<i>Iniciar Processo com dados de formulário</i>	49
<i>Módulo de exportação de Dados</i>	52

Exemplo de utilização --- 55

Introdução

Este documento descreve a APIs REST do Zuri.

Sobre a nova versão da API

A partir da versão 8.14 o Zuri passou a expor um novo conjunto de API beta para aplicações externas e mobile. Os métodos de API REST do Zuri existentes nas versões anteriores continuarão sendo suportados até que as novas APIs sejam lançadas em versão final.

Abaixo segue uma lista dos endpoints da API atual e seus equivalentes na nova API. As diferenças entre os endpoints atuais e os novos estarão descritas na documentação de cada novo item.

Endpoint Atual	Enpoint Novo	Comentário
/ZuriServices/Authenticate	/api/auth/	Autentica o usuário e devolve um Token
/ZuriServices/ListProcess/	/api/zuri/procs/	Lista de processos
/ZuriServices/ListInbox/	/api/zuri/items/inbox/	Lista de itens na caixa de entrada do usuário
/ZuriServices/GetProcessItemStepInfo	/api/zuri/items/{id}/{step}/	Detalhes de item na Caixa de Entrada. Esta url agora é retornada como parte do objeto obtido na listagem da caixa de entrada
/ZuriServices/ExecuteWorkflow	/api/zuri/items/{id}/{step}/exec/	Executa uma operação em um item

Definições

As APIs descritas neste documento estão preparadas para trabalhar com objetos formatados seguindo a notação JSON, tanto nos parâmetros enviados para os métodos como nos seus resultados.

Todos os valores numéricos recebidos ou devolvidos pelos serviços deverão estar formatados com cultura neutra (sem separador de milhar e usando "." [ponto] como deparador decimal).

Todos os valores em formato data deverão estar em formato ISO, conforme exemplo a seguir: "2014-07-31T14:40:31.3349328-03:00" (dia 31 de julho de 2014, às 14:40:31, horário de Brasília).

Abaixo segue uma descrição da estrutura de todos os objetos esperados ou retornados pelas APIs aqui definidas.

Zuri Settings

Este objeto representa as configurações do Zuri a serem utilizadas nos aplicativos mobile.

Nome do campo	Tipo	Descrição
theme	theme[]	Array de objetos do tipo theme, contendo o nome, tipo e valor. O tipo define se o valor é uma cor ou url de imagem.
menus	menu[]	Array de objetos do tipo menu, contendo nome, tipo de menu e uma url que para cada tipo de menu tem um significado diferente.

Exemplo em JSON

```
{
  "menus":[
    {
      "name":"Processos",
      "type": "Processes"
      "url": "/api/zuri/procs/"
    },
    {
      "name":"Pontos de Interesse",
      "type": "Map"
      "url": "/api/zuri/map/default/{coord}/"
    },
    {
      "name":"Visitas Agendadas",
      "type": "Inbox"
      "url": "/api/zuri/items/inbox/?procid=1"
    }
  ],
  "theme":[
    {
      "name":"Theme",
      "type":"Color",
      "value": "#12364A"
    },
    {
      "name":"Contrast",
      "type":"Color",
      "value": "#FFFFFF"
    },
    {
      "name":"Background",
      "type":"Color",
      "value": "#CCCCCC"
    }
  ]
}
```

```

    },
    {
      "name": "Text",
      "type": "Color",
      "value": "#444444"
    },
    {
      "name": "Logo",
      "type": "Color",
      "url": "/WFResources/Themes/Images/zuri-admin.png",
      "value": "data:image/png;base64,asdfasfdas..."
    }
  ]
}

```

Theme

Este objeto representa uma das cores de personalização do aplicativo.

Nome do campo	Tipo	Descrição
name	string	Nome da url de conteúdo.
type	string	Indica se é uma cor ou imagem
value	string	Cor em formato RGB hexadecimal usado em HTML ou url da image. Se for imagem esse campo contém uma string formatada em base64 com a imagem do logo
url	string	Presente apenas para tipo imagem. Contem a url da imagem.

Exemplo em JSON

```

{
  "name": "Background",
  "type": "Color",
  "value": "#FF0000"
}

```

Menu

Este objeto representa um item de menu do aplicativo. Os tipos de menu disponíveis estão na tabela de [tipos de menu](#) mais adiante neste documento.

Nome do campo	Tipo	Descrição
name	string	Nome do menu.
type	tipo de menu	Tipo de menu
url	string	Url do menu. Para cada tipo de menu essa url é usada de uma forma diferente.

Exemplo em JSON

```
{
  "name": "Processos",
  "type": "Processes",
  "url": "/api/zuri/procs/"
}
```

Ponto no Mapa

Este objeto representa um ponto em um mapa.

Nome do campo	Tipo	Descrição
latitude	double	Latitude da coordenada do ponto no mapa, com "." Como separador de decimal.
longitude	double	Longitude da coordenada do ponto no mapa, com "." Como separador de decimal.
name	string	Url do menu. Para cada tipo de menu essa url é usada de uma forma diferente.
description	string	Opcional. Descrição do local no mapa
url	string	Opcional. Url de webview para visualização de detalhes do ponto no mapa.
image	string	Opcional. Url de ícone a ser exibido no no balão de detalhes do ponto no mapa

html	string	Opcional. Html de detalhes do ponto no mapa. Este campo deve ser ignorado se o campo url tiver sido informado.
------	--------	--

Exemplo em JSON

```
{
  "latitude": "-23.564620",
  "longitude": "-46.690923",
  "name": "Web Foundations",
  "description": "Escritório matriz da WF",
  "url": "http://solucoes.wf",
  "image": "",
  "html": ""
}
```

Processo

Este objeto representa um Processo.

Nome do campo	Tipo	Descrição
name	string	Nome do processo
inboxCount	int	Total de itens deste processo na caixa de entrada do usuário
startUrl	string	Url do endpoint que deve ser chamado para iniciar um item desse tipo. Se este parâmetro estiver vazio significa que o usuário não tem permissão para iniciar itens desse processo. Este campo está obsoleto (mas continuará sendo retornado) já que agora as urls vêm em um array na propriedade "links".
inboxUrl	string	Url do endpoint que deve ser chamado para listar os itens desse processo na caixa de entrada do usuário. Este campo está obsoleto (mas continuará sendo retornado) já que agora as urls vêm em um array na propriedade "links".
links	Link []	Coleção de links de ações na API relacionados ao processo. Link do tipo "create-item" contem a mesma url indicada na propriedade "startUrl". Link do tipo "items-inbox" contém a mesma url indicada na propriedade "inboxUrl". Os links de tipo "items-mine" e "items-following" representam respectivamente as listas de meus itens e acompanhamento do usuário

Exemplo em JSON

```
{
  "name": "Agendamento de Visita",
  "inboxCount": 2,
  "startUrl": "/api/zuri/procs/1/items/",
  "inboxUrl": "/api/zuri/items/inbox/?procid=1",
  "links": [
    {
      "rel": "create-item",
      "title": "Caixa de entrada",
      "href": "/api/zuri/procs/1/items/"
    }, {
      "rel": "items-inbox",
      "title": "Caixa de entrada",
      "href": "/api/zuri/items/inbox/?procid=1"
    }, {
      "rel": "items-mine",
      "title": "Meus itens",
      "href": "/api/zuri/items/mine/?procid=1"
    }, {
      "rel": "items-following",
      "title": "Acompanhamento",
      "href": "/api/zuri/items/following/?procid=1"
    }
  ]
}
```

Link

Representa um link para outro recurso dentro da API.

Nome do campo	Tipo	Descrição
rel	string	Tipo do link.
title	string	Nome do link
href	string	Url do link

Item na caixa de entrada, meus itens ou acompanhamento

Este objeto representa um item na caixa de entrada, meus itens ou lista de acompanhamento do usuário.

Nome do campo	Tipo	Descrição
id	int	Id do item no Zuri
name	string	Nome do item
status	string	Status do item
step	string	Etapa em que esse item está
stepInfo	String	Opcional. Informação extra da etapa, configurada no campo “Identificador” da aba “Configurações Avançadas” da Etapa.
info	string	Opcional. Informação extra a ser exibida sobre o item
detailsUrl	string	Url de endpoint que deve ser usada para obter os detalhes de execução do item. Se estiver vazio significa que não há detalhes extras para exibir sobre o item para o usuário atual.
stepSLA	SLA	SLA desta etapa
itemSLA	SLA	SLA deste item
gps	GPS []	Array contendo todos os GPSs desse item de processo que o usuário tiver acesso. Cada GPS por sua vez contém um array de links para suas versões em JPG e SVG. Se este campo estiver vazio ou não for retornado pela API significa que o processo não tem GPS configurado ou o usuário atual não tem acesso a nenhum dos GPS
assignedTo	string	Nome do usuário ao qual o item está vinculado ou “N/A” se não estiver vinculado a ninguém.

Exemplo em JSON

```
{
  "id": "1",
  "name": "Reembolso de Taxi",
  "status": "Aguardando Aprovação",
  "step": "Aprovação",
```

```
{
  "info": "Valor R$ 110,00",
  "assignedTo": "Fulano de Tal",
  "detailsUrl": "/api/zuri/items/6/12/",
  "stepSLA":
  {
    "name": "No prazo",
    "color": "#049D01",
    "isDelay": false,
    "deadline": null
  },
  "itemSLA":
  {
    "name": "No prazo",
    "color": "#049D01",
    "isDelay": false,
    "deadline": "2019-02-25T00:00:00"
  }
}
```

SLA

Informação de SLA referente a um item ou etapa.

Nome do campo	Tipo	Descrição
name	string	Nome do SLA
color	string	Representação Hexadecimal em formato HTML da cor definida para este SLA
isDelay	boolean	Indica se este SLA representa um atraso
deadline	date	Indica a data limite de execução para esta etapa/item ser considerado atrasado.

GPS

Este objeto representa uma visualização de GPS disponível no item atual para o usuário corrente. Um processo pode ter mais de uma visualização de GPS configuradas, com definições diferentes e para diferentes grupos de usuários.

Nome do campo	Tipo	Descrição
---------------	------	-----------

name	string	Nome do GPS
links	GPS Resource []	Lista de URLs disponíveis para esse GPS. Cada URL representa um formato diferente para o GPS (imagem JPEG ou SVG). Cada aplicativo deverá usar o formato que fizer mais sentido e proporcionar a melhor experiência para o usuário.

GPS Resource

Representa um link para um tipo de GPS em um formato específico. Cada GPS pode ser exibido em mais de um formato (JPEG ou SVG), portanto terá um link distinto para cada formato suportado.

Nome do campo	Tipo	Descrição
type	string	Content-type do GPS. O GPS pode ser representado por uma imagem estática (JPG) ou uma representação vetorial (SVG). Se for uma imagem, o type será "image/jpeg" . Se for vetorial o type será "image/svg+xml" .
url	string	Url do GPS. Chamar essa url retorna uma imagem JPG ou SVG desse gps do item. Essa url suporta o verbo GET no request e não requer autenticação.

Ítem

Este objeto representa um ítem em processamento.

Nome do campo	Tipo	Descrição
id	int	Id do ítem no Zuri
activityId	int	Informação utilizada internamente pelo Zuri
name	string	Nome do ítem

assignedTo	string	Nome do usuário ao qual o item está vinculado ou “N/A” se não estiver vinculado a ninguém.
gps	GPS[]	Array contendo todos os GPSs desse processo que o usuário tiver acesso. Cada GPS por sua vez contém um array de links para suas versões em JPG e SVG. Se este campo estiver vazio ou não for retornado pela API significa que o processo não tem GPS configurado ou o usuário atual não tem acesso a nenhum dos GPS
execUrl	string	Url de endpoint que deverá ser usado para execução do item. Se estiver vazio significa que o usuário atual não pode executar este item (tomar decisão ou preencher formulário)
execMessage	string	Mensagem de execução retornada pela api
stepSLA	SLA	SLA desta etapa
itemSLA	SLA	SLA deste item
execType	string	Tipo de execução retornado. É usado para identificar qual tipo de tela será apresentada (decisão, formulário...)
execStatus	int	Estado de execução retornado. É usado para saber se deve ser apresentada alguma tela, mensagem de erro ou se o processamento acabou e deve voltar para a listagem.
execData	Dados execução	O tipo desse parametro depende do execType, quando retornado pela API.

Exemplo em JSON

```
{
  "id": 1,
  "activityId": 2,
  "name": "Reembolso de Taxi",
  "assignedTo": "Fulano de Tal",
  "execUrl": "/api/zuri/items/6/12/exec/",
  "execMessage": "",
  "execType": "b4aa1704-c8b5-46ae-8e70-ce00253ee023",
  "execStatus": 0,
  "execData": {
    "message": "Deseja aprovar o reembolso de fulado no valor de R$ 110,00",
    "decisions" : [{ "id": 4, "name": "Aprovar" }, { "id": 5, "name": "Reprovar" } ]
  }
}
```

```

    }/”,
    "stepSLA":
    {
        "name": "No prazo",
        "color": "#049D01",
        "isDelay": false ,
        "deadline": null
    },
    "itemSLA":
    {
        "name": "No prazo",
        "color": "#049D01",
        "isDelay": false,
        "deadline": null
    }
}

```

Dados de Execução

Este objeto representa os detalhes de execução de um item. Quando retornado pela api o conteúdo desse objeto depende do execType retornado no detalhe do Ítem.

Abaixo há uma tabela contendo a lista de tipos de execução existentes e seus respectivos formatos para objetos retornados pela API. A api está padronizada para retornar esses códigos sempre em caixa baixa (letras minúsculas), mas mesmo assim recomenda-se que a checagem na aplicação seja feita de forma case-insensitive.

execType (API > mobile app)	Tipo	Descrição
b4aa1704-c8b5-46ae-8e70-ce00253ee023	Tomada de Decisão	Detalhes para tela de tomada de decisão
1b34d0fa-d951-425a-97f7-7f1c6652794d	Definição de formulário	Detalhes para tela de preenchimento de formulário

Essa mesma propriedade deve ser usada pelo aplicativo mobile para enviar os dados de execução de volta para a api.

Abaixo há uma tabela contendo a lista de tipos de execução existentes e seus respectivos formatos para objetos enviado pelo aplicativo para a API.

execType (mobile app > API)	Tipo	Descrição
b4aa1704-c8b5-46ae-8e70-ce00253ee023	Decisão	Decisão que foi tomada pelo usuário.

Tomada de Decisão

Este objeto representa os detalhes de execução para uma tomada de decisão. Se não retornar a lista de decisões, significa que o item não está na caixa de entrada do usuário, mas está na lista de meus itens ou acompanhamento. Neste caso o usuário pode ver, mas não pode tomar decisões.

Nome do campo	Tipo	Descrição
message	string	Mensagem a ser exibida na tela de tomada de decisão
decisions	decisão []	Array de decisões disponíveis para esta etapa. Se não houver decisões nesta lista, significa que o usuário atual não pode tomar decisões nesta etapa, mas pode ver o item pois está na lista de acompanhamento (observadores) ou meus itens.

Exemplo em JSON

```
{
  "message": "Deseja aprovar o reembolso de fulano no valor de R$ 110,00?",
  "decisions": [{"id":4, "name":"Aprovar"}, {"id":5, "name":"Reprovar"}]
}
```

Decisão

Este objeto representa uma decisão.

Nome do campo	Tipo	Descrição
id	int	Id da decisão
name	string	Nome da decisão

Exemplo em JSON

```
{
  "id": "1",
  "name": "Aprovar"
}
```


Definição de formulário

Este objeto representa as definições de um formulário a ser preenchido pelo usuário.

Nome do campo	Tipo	Descrição
Id	int	Id do formulário
name	string	Nome do formulário
dataSourcesUrl	string	Url que deverá ser chamada se o formulário tiver campos com dependência e um desses campos for alterado.
fields	campo de formulário []	Array contendo as definições dos campos do formulário.

Exemplo em JSON

```
{
  "id": 1,
  "name": "Dados da Visita",
  "dataSourcesUrl": "/api/ninki/forms/1/items/1/sources/"
  "fields": [campos de formulário]
}
```

Campo de Formulário

Este objeto representa as definições de um campo de formulário. Abaixo estão as definições gerais e específicas de cada tipo de campo de formulário existente.

As propriedades abaixo são comuns à todos os tipos de campos.

Nome do campo	Tipo	Descrição
name	string	Nome do campo (para uso interno)
title	string	Título do campo

isRequired	boolean	Indica se o campo é obrigatório. Não se aplica ao campo tipo group . O formulário não deve ser enviado com campos obrigatórios sem preenchimento.
isReadOnly	boolean	Indica se o campo pode ser preenchido na interface ou se seu valor deve ser apenas apresentado.
updateSources	boolean	Indica se alterações neste campo devem causar uma atualização nos data sources do formulário. Caso algum campo que esteja marcado com “true” seja alterado, a rotina de atualização de sources do formulário deverá ser executada (isso não se aplica a preenchimentos de formulário offline). Esta propriedade não se aplica a campos do tipo group
groupName	string	Define se o campo faz parte de algum agrupador e deve ser mostrado “collapsed”. O único campo que não tem esta propriedade é o Action pois não podem haver botões dentro de um grupo
type	Tipo de campo	Tipo do campo. Ver os detalhes de cada tipo para descobrir suas propriedades adicionais.
validationGroup	string	Define o grupo de campos relacionados para validação. Se for uma Action define quais controle valida ao clicar. Por exemplo ao clicar em uma Action com o validation group “xpto” todos os campos com validation group “xpto” deverão ser validados na tela, os outros deverão ser ignorados. Quando uma Action não tem nenhum valor em validationgroup ela valida todos os campos da tela sempre.

Campo Action

Representa um botão de ação a ser exibido na tela. Ao ser clicado, o formulário deverá ser validado e se estiver ok, deverá ser enviado para a url de execução do item.

Obs. O campo do tipo Action não pode ser usado em agrupador

Nome do campo	Tipo	Descrição
value	string	Valor da opção selecionada

Campo Boolean

Checkbox ou equivalente.

Nome do campo	Tipo	Descrição
value	boolean	Valor inicial do campo.

Campo Date

Campo de data sem hora.

Nome do campo	Tipo	Descrição
value	date	Valor inicial do campo (yyyy-MM-ddTHH:mm:ss).

Campo DateTime

Campo de data com hora

Nome do campo	Tipo	Descrição
value	datetime	Valor inicial do campo (yyyy-MM-ddTHH:mm:ss).

Campo Time

Campo de hora sem data.

Nome do campo	Tipo	Descrição
value	time	Valor inicial do campo (HH:mm:ss).

Campo DateTimeTime

Campo de data e hora com hora final (para definição de intervalos de tempo em um dia, com uma reunião)

Nome do campo	Tipo	Descrição
value	datetime	Valor inicial do campo. Contém a data e hora iniciais (yyyy-MM-ddTHH:mm:ss).
finishTime	time	Hora final (HH:mm:ss)

finishTimeOffset	int	Se definido faz com que o valor para o fim seja colocado automaticamente depois de definido o início adicionando X horas
------------------	-----	--

Campo Label

Campo de mensagem de texto para exibição. Este campo não contém nenhuma propriedade adicional. O título do campo está na propriedade title e o seu conteúdo na propriedade value.

Campo Position

Campo para obtenção da posição de GPS do usuário.

Nome do campo	Tipo	Descrição
latitude	double	Latitude da posição do GPS no momento do preenchimento do formulário.
longitude	double	Longitude da posição do GPS no momento do preenchimento do formulário.
range	double	Opcional. Se informado e maior que 0, indica a distância em metros que o aplicativo deve monitorar a posição do usuário. Se o usuário se distanciar além do range informado, o aplicativo deverá executar o procedimento de atualização de dados de formulário , usando a url que está na propriedade RangeUrl da definição deste campo.
rangeURL	string	Opcional. url que deverá ser usada para notificar que o usuário saiu do range definido pelo campo.
startTime	datetime	Data e hora em que chegou na localidade (yyyy-MM-ddTHH:mm:ss).
exitTime	datetime	Opcional. Data e hora de quando ocorreu a saída (yyyy-MM-ddTHH:mm:ss).

Campo Map

Ponto no mapa a ser exibido no aplicativo.

Nome do campo	Tipo	Descrição
latitude	double	Latitude da posição que deverá ser exibida no mapa

longitude	double	Longitude da posição que deverá ser exibida no mapa
height	int	Altura em pixels do mapa na tela
locName	string	Título do ponto localizado no mapa
locDescription	string	Informações adicionais (segunda linha abaixo do título na tag marcada o mapa)
showMap	boolean	Define se o mapa deve ser mostrado na tela ou somente o traçar rota
showRoute	boolean	Define se tem que mostrar o traçar rota

Campo Scale

Slider/Termometro. O valor desse campo deve ser um decimal entre 0 e 1.

Nome do campo	Tipo	Descrição
value	integer	Valor inicial do campo.
lowValue	integer	Primeiro valor da escala
highValue	integer	Último valor da escala

Campo Rating

Classificação 5 estrelas. O valor desse campo deve ser um decimal entre 0 e 1.

Nome do campo	Tipo	Descrição
Value	integer	Valor inicial do campo. 0 a 5

Campo Option

Controle dropdown contendo diversas opções para escolha (ex.: Cidades).

Nome do campo	Tipo	Descrição
value	string	Valor da opção selecionada
items	Item de opções[]	Array das opções disponíveis no controle, contendo name/value.

Campo Multiple

Lista contendo diversas opções para múltipla escolha.

Nome do campo	Tipo	Descrição
value	string	Valor das opções selecionadas, separadas por “,” (vírgula)
items	Item de opções[]	Array das opções disponíveis no controle, contendo name/value.

Campo Text

Campos de texto simples.

Nome do campo	Tipo	Descrição
value	string	Valor inicial do campo
maxLength	int	Tamanho máximo do campo
lineCount	integer	Número de linhas disponibilizadas para digitação (define a altura do campo)
isPassword	boolean	Se for true trata-se de um campo para a digitação de uma senha
validationExpression	string	Opcional. Expressão regular de validação do campo. Se for informada o campo só será considerado válido se seu conteúdo estiver de acordo com a expressão passada. Se o campo estiver marcado com não obrigatório mas tiver uma expressão regular, esta só deverá ser validada se o campo não estiver vazio.
mask	string	Opcional. Máscara para digitação do campo (por exemplo 99999-999 para CEP)
watermark	string	Opcional. Define o texto que aparece antes da digitação

Campo HiddenValue

Este campo não monta nada na tela, ele só precisa ser recebido e postado de volta

Nome do campo	Tipo	Descrição
value	string	Valor do campo

Campo Group

Representa um agrupamento de campos na tela. O texto em title deve ser exibido como título do agrupamento.

Nome do campo	Tipo	Descrição
fields	campos de formulário []	Array dos campos existentes no agrupamento.

Exemplo em JSON contendo todos os campos de formulário listados acima

```
{
  "id": 1,
  "name": "Dados da Visita",
  "dataSourcesUrl": "/api/ninki/forms/1/items/1/sources/",
  "fields": [
    {
      "name": "campo_01",
      "updateSources": false,
      "isRequired": true,
      "title": "Nome",
      "value": "",
      "maxLength": 100,
      "validationExpression": "",
      "type": "text"
    },
    {
      "name": "campo_02",
      "updateSources": false,
      "isRequired": false,
      "title": "Quantidade",
      "value": 0,
      "validationExpression": "",
      "type": "int"
    },
    {
      "name": "campo_03",
      "updateSources": false,
      "isRequired": false,
      "title": "Preço",
      "value": 0,
      "validationExpression": "",
      "type": "decimal"
    },
    {
      "name": "campo_04",
      "updateSources": false,
      "isRequired": false,
      "title": "Deseja receber e-mail?",
      "value": false,
      "type": "boolean"
    },
    {
      "name": "campo_05",
      "updateSources": false,
      "isRequired": false,
      "title": "Nota do Atendimento",
      "value": "",
      "type": "scale"
    },
    {
      "name": "campo_06",
      "updateSources": false,
      "isRequired": false,
      "title": "Classificação",
      "value": "",
      "type": "rating"
    },
    {
      "name": "campo_07",
      "updateSources": false,
      "isRequired": false,
      "title": "Dia",
      "value": "",
      "type": "date"
    },
    {
      "name": "campo_08",
      "updateSources": false,
      "isRequired": false,
      "title": "Dia e Hora",
      "value": "",
      "type": "datetime"
    },
    {
      "name": "campo_09",
      "updateSources": false,
      "isRequired": false,
      "title": "Período Entrevista",
      "value": "",
      "finishTime": "",
      "type": "datetimetime"
    },
    {
      "name": "campo_10",
      "updateSources": false,
      "isRequired": false,
      "title": "Esta é uma mensagem qualquer do formulário",
      "type": "label"
    }
  ]
}
```

```

{
  "name": "campo_11", "updateSources": false, "isRequired": false,
  "title": "CheckIn", "latitude": "", "longitude": "", "range": 0,
  "exitTime": "", "type": "position"},
  "name": "campo_12", "updateSources": false, "isRequired": false,
  "title": "Local da Loja", "latitude": -23.56462,
  "longitude": -46.690923, "type": "map"},
  "name": "campo_13", "updateSources": false, "isRequired": false,
  "title": "Estado", "value": "", "type": "option", "items": [
    { "name": "São Paulo", "value": "SP" },
    { "name": "Rio de Janeiro", "value": "RJ" }
  ]},
  "name": "campo_14", "updateSources": false, "isRequired": false,
  "title": "Participantes", "value": "", "type": "multiple", "items": [
    { "name": "Fulado da Silva", "value": "1" },
    { "name": "Bruce Wayne", "value": "21" },
    { "name": "Beltrano dos Santos", "value": "15" }
  ]},
  "name": "campo_15", "updateSources": false, "isRequired": false,
  "title": "Ok", "value": "ok", "type": "action"},
  "name": "campo_16", "title": "Informações Complementares",
  "type": "group", "fields": [
    { "name": "campo_17", "updateSources": false, "isRequired": false,
      "title": "Nome", "value": "", "maxLength": 100,
      "validationExpression": "", "type": "text" },
    { "name": "campo_18", "updateSources": false, "isRequired": false,
      "title": "Quantidade", "value": 0,
      "validationExpression": "", "type": "int" }
  ]
}

```

Item de campo de opções

Representa um item de uma lista de opções usadas nos campos de tipo “option” e “multiple”.

Nome do campo	Tipo	Descrição
name	string	Nome da opção
value	string	Valor da opção

Sources de formulário

Este objeto contém informações referentes às fontes de dados de um formulário. Ele é obtido há dependências entre campos de formulário e alterações em um campo disparam uma chamada para obter fontes de dados. Exemplo: Se for retornado na propriedade fields um campo de formulário com nome “campo10” e propriedade items contendo um array de (name,value) significa que a propriedade items do campo10 do formulário atual deverá ser atualizada para a lista de itens informada.

Nome do campo	Tipo	Descrição
---------------	------	-----------

fields	campo de formulário []	Array contendo todos os campos do formulário atual que devem ser atualizados.
--------	--	---

Dados de formulário

O objeto de dados de formulário deve ser um objeto composto de uma propriedade para cada campo do formulário, seguida de seu respectivo valor. Exemplo: Para um formulário que tem 2 campos tipo texto chamados “campo1” e “campo2” o objeto de dados se formulário seria semelhante ao definido abaixo.

Nome do campo	Tipo	Descrição
trigger	string	Nome do controle responsável pela ação. Se a ação for uma execução de fluxo essa propriedade deve conter o nome do botão tipo “action” que foi clicado.
data	valores de campo	Objeto contendo uma propriedade para cada campo do formulário. Cada tipo de campo gera uma propriedade com formato diferente.

Exemplo do objeto de dados de formulário em JSON para um formulário semelhante ao exemplo de definição de formulário acima.

```
{
  "trigger": "campo_15",
  "data": {
    "campo_01": "nome de alguma coisa",
    "campo_02": 1,
    "campo_03": 10.32,
    "campo_04": false,
    "campo_05": 0.6,
    "campo_06": 0.8,
    "campo_07": "2014-12-01",
    "campo_08": "2014-12-01T15:50.000-03:00",
    "campo_09": {value:"2014-12-01T15:50.000-03:00", finishTime: "16:50"},
    "campo_11": {latitude:-23.000, longitude:-46.000},
    "campo_13": "SP",
    "campo_14": "1,21",
    "campo_15": "ok",
    "campo_17": "outro campo texto",
    "campo_18": 4
  }
}
```

Valores de Campo

Objeto contendo uma propriedade para cada campo do formulário preenchido. Cada tipo de campo gera uma propriedade com formato diferente. Abaixo seguem as explicações para cada tipo de campo.

Campos dos tipos “label”, “map” e “group” não retornam nada no objeto de dados de formulário.

Campo Action

Retorna uma propriedade cujo nome é o name do campo e o valor é o value do botão que foi clicado. Pode haver mais de 1 campo do tipo action com o mesmo name no formulário, mas no objeto de dados do formulário deve haver apenas uma propriedade e o valor deve sempre conter o valor do campo action que foi clicado ou ser vazio caso nenhum action tenha sido clicado (no caso de um objeto sendo enviado para solicitação de atualização de sources). No exemplo JSON acima o “campo_15” é do tipo action.

Campo Text

Retorna uma propriedade cujo nome é o name do campo e o valor é o texto que foi preenchido pelo usuário. No exemplo JSON acima “campo_01” e “campo_17” são do tipo text.

Campo Int

Retorna uma propriedade cujo nome é o name do campo e valor é o número inteiro que foi preenchido pelo usuário. No exemplo JSON acima “campo_02” e “campo_18” são do tipo int.

Campo Decimal

Retorna uma propriedade cujo nome é o name do campo e valor é o número decimal que foi preenchido pelo usuário. No exemplo JSON acima o “campo_03” é do tipo decimal.

Campo Boolean

Retorna uma propriedade cujo nome é o name do campo e valor é um boolean (true|false) indicando se o usuário marcou o checkbox. No exemplo JSON acima o “campo_04” é do tipo boolean.

Campo Scale

Retorna uma propriedade cujo nome é o name do campo e valor é um número decimal cujo mínimo e máximo é passado junto com a definição do campo. A apresentação deste campo deve ser algo no formato slider. No exemplo JSON acima o “campo_05” é do tipo scale.

Campo Rating

Retorna uma propriedade cujo nome é o name do campo e valor é o número decimal entre 0 e 5 referente ao número de estrelas selecionados. No exemplo JSON acima o “campo_06” é do tipo rating.

Campo Date

Retorna uma propriedade cujo nome é o name do campo e valor é a data em formato “yyyy-MM-dd” que foi preenchida pelo usuário. No exemplo JSON acima o “campo_07” é do tipo date.

Campo Datetime

Retorna uma propriedade cujo nome é o nome do campo e valor é a data e hora em formato ISO que foi preenchido pelo usuário. Ex.: 2014-07-31T14:40:31:33. No exemplo JSON acima o “campo_08” é do tipo datetime.

Campo Datetime

Retorna uma propriedade cujo nome é o nome do campo e valor é um objeto com as propriedades value contendo a data e hora de início em formato ISO que foi preenchido pelo usuário e a propriedade finishTime contém a hora de fim. No exemplo JSON acima o “campo_09” é do tipo datetime.

Campo Option

Retorna uma propriedade cujo nome é o nome do campo e valor é a propriedade value da da opção escolhida pelo usuário. No exemplo JSON acima o “campo_13” é do tipo option.

Campo Multiple

Retorna uma propriedade cujo nome é o nome do campo e valor é um texto contendo os valores das propriedades value das opções escolhidas pelo usuário separadas por “,”. No exemplo JSON acima o “campo_14” é do tipo multiple.

Campo Position

Retorna uma propriedade cujo nome é o nome do campo e valor é um objeto com as propriedades latitude e longitude das coordenadas de GPS do usuário no momento do preenchimento do formulário. No exemplo JSON acima o “campo_11” é do tipo position.

Item Detalhado

Objeto que possui os dados de histórico do item de processo.

Nome do campo	Tipo	Descrição
id	int	Id do item.
nome	string	Nome do item.
status	string	Status do item.
dataInicio	datetime	Data de início do item.
dataFim	datetime	Data de finalização do item. Caso esse campo esteja preenchido, significa que o item já está finalizado.
readOnly	boolean	Somente leitura.

usuOwner	int	Dono do item.
areaOrigem	int	Id da área de origem.
itemOrigem	int	Id do item de origem.
processoOrigem	int	id do processo de origem.
param1	string	Parâmetro opcional 1.
param2	string	Parâmetro opcional 2.
param3	string	Parâmetro opcional 3.
param4	string	Parâmetro opcional 4.
param5	string	Parâmetro opcional 5.
historico	Item Historico	Lista de histórico do item.

Exemplo do objeto de detalhes do histórico de um item de processo:

```
{
  "id": 15,
  "nome": "Desenvolvedor C#",
  "status": "Inicializado",
  "dataInicio": "2017-08-07T10:55:23",
  "dataFim": "2017-08-07T11:11:25",
  "readOnly": false,
  "usuOwner": 6,
  "areaOrigem": 1,
  "itemOrigem": 0,
  "processoOrigem": 4,
  "param1": "Aprovado",
  "param2": "5",
  "param3": "2",
  "param4": "Jorge dos Santos",
  "param5": "Desenvolvedor C#",
  "historico": [
    {
      "executor": "João da Silva",
      "execucao": "07/08/2017 10:55",
      "etapa": "Inicio",
      "descricao": "Iniciou a etapa Inicio",
      "status": "Inicializado",
      "acao": "Etapa",
      "tipo": 2,

```

```
    "subProcesso": ""
  },
  {
    "executor": "Maria Lourdes da Silva",
    "execucao": "07/08/2017 10:55",
    "etapa": "Inicio",
    "descricao": "Finalizou a etapa Inicio",
    "status": "Inicializado",
    "acao": "N/A",
    "tipo": 22,
    "subProcesso": ""
  },
  {
    "executor": "João da Silva",
    "execucao": "07/08/2017 10:55",
    "etapa": "Inicio",
    "descricao": "Formulario - dados da vaga preenchido.",
    "status": "Inicializado",
    "acao": "Formulario Web",
    "tipo": 3,
    "subProcesso": ""
  },
  {
    "executor": "João da Silva",
    "execucao": "07/08/2017 10:55",
    "etapa": "Incluir o Candidato",
    "descricao": "Iniciou a etapa Incluir o Candidato",
    "status": "Inicializado",
    "acao": "Etapa",
    "tipo": 2,
    "subProcesso": ""
  },
  {
    "executor": "Maria Lourdes da Silva",
    "execucao": "07/08/2017 10:55",
    "etapa": "Incluir o Candidato",
    "descricao": "Incluir",
    "status": "Inicializado",
    "acao": "Decisão",
    "tipo": 1,
    "subProcesso": ""
  },
  {
    "executor": "Maria Lourdes da Silva",
    "execucao": "07/08/2017 10:55",
    "etapa": "Incluir o Candidato",
    "descricao": "Análise de Candidato [9] iniciado.",
    "status": "Inicializado",
    "acao": "Subprocesso Início",
    "tipo": 3,
    "subProcesso": "Análise de Candidato"
  },
  {
    "executor": "Maria Lourdes da Silva",
    "execucao": "07/08/2017 10:56",
    "etapa": "Avaliar Contratação",
    "descricao": "Iniciou a etapa Análise de Candidato - Avaliar Contratação",
```

```
    "status": "Inicializado",
    "acao": "Etapa",
    "tipo": 2,
    "subProcesso": "Análise de Candidato"
  },
  {
    "executor": "Lorena Silva",
    "execucao": "07/08/2017 10:58",
    "etapa": "ENTRADA",
    "descricao": "Iniciou a etapa ENTRADA",
    "status": "Inicializado",
    "acao": "Etapa",
    "tipo": 2,
    "subProcesso": ""
  },
  {
    "executor": "Lorena Silva",
    "execucao": "07/08/2017 10:58",
    "etapa": "ENTRADA",
    "descricao": "Finalizou a etapa ENTRADA",
    "status": "Inicializado",
    "acao": "N/A",
    "tipo": 22,
    "subProcesso": ""
  },
  {
    "executor": "Administrador",
    "execucao": "07/08/2017 11:09",
    "etapa": "Criação dos acessos",
    "descricao": "Fornecer dados de acesso",
    "status": "Inicializado",
    "acao": "Decisão",
    "tipo": 1,
    "subProcesso": ""
  },
  {
    "executor": "Administrador",
    "execucao": "07/08/2017 11:10",
    "etapa": "Formalizar Contratação",
    "descricao": "Iniciou a etapa Formalizar Contratação",
    "status": "Inicializado",
    "acao": "Etapa",
    "tipo": 2,
    "subProcesso": ""
  },
  {
    "executor": "Administrador",
    "execucao": "07/08/2017 11:10",
    "etapa": "Criação dos acessos",
    "descricao": "Solicitação de Contratação - Infra finalizado",
    "status": "Inicializado",
    "acao": "N/A",
    "tipo": 9,
    "subProcesso": ""
  },
  {
    "executor": "Administrador",
```

```

        "execucao": "07/08/2017 11:10",
        "etapa": "Criação dos acessos",
        "descricao": "Finalizou a etapa Criação dos acessos",
        "status": "Inicializado",
        "acao": "N/A",
        "tipo": 22,
        "subProcesso": ""
    },
    {
        "executor": "Administrador",
        "execucao": "07/08/2017 11:10",
        "etapa": "Criação dos acessos",
        "descricao": "Formulario ctt - contratação preenchido.",
        "status": "Inicializado",
        "acao": "Formulario Web",
        "tipo": 3,
        "subProcesso": ""
    },
    {
        "executor": "Administrador",
        "execucao": "07/08/2017 11:10",
        "etapa": "Criação dos acessos",
        "descricao": "Fluxo externo chamado Solicitação de Contratação [Retorno
Infra]",
        "status": "Inicializado",
        "acao": "Fluxo externo",
        "tipo": 3,
        "subProcesso": ""
    },
    {
        "executor": "Administrador",
        "execucao": "07/08/2017 11:10",
        "etapa": "Retorno Infra",
        "descricao": "Iniciou a etapa Retorno Infra",
        "status": "Inicializado",
        "acao": "Etapa",
        "tipo": 2,
        "subProcesso": ""
    }
    {
        "executor": "Arthur Marins",
        "execucao": "07/08/2017 11:11",
        "etapa": "Formalizar Contratação",
        "descricao": "Finalizou a etapa Formalizar Contratação",
        "status": "Inicializado",
        "acao": "N/A",
        "tipo": 22,
        "subProcesso": ""
    },
    {
        "executor": "Arthur Marins",
        "execucao": "07/08/2017 11:11",
        "etapa": "Formalizar Contratação",
        "descricao": "Email enviado: Processo de contratação concluído
[arthur@zuri.wf;]",
        "status": "Inicializado",
        "acao": "E-mail",

```

```
        "tipo": 3,  
        "subProcesso": ""  
    }  
]  
}
```

Item Histórico

Objeto com os detalhes do histórico do item.

Nome do campo	Tipo	Descrição
executor	string	Nome do executor.
execução	string	Data de execução em texto no formato “dd/mm/yyyy hh:mm”
etapa	string	Descrição da etapa.
descrição	string	Descrição do histórico.
status	string	Status do histórico.
acao	string	Tipo da ação.
tipo	int	Enumerador do tipo de histórico.
subProcesso	string	Descrição do subprocesso.

Segue abaixo exemplo de um objeto do tipo de histórico:

```
{  
  "executor": "Arthur Marins",  
  "execucao": "07/08/2017 10:55",  
  "etapa": "Inicio",  
  "descricao": "Iniciou a etapa Inicio",  
  "status": "Inicializado",  
  "acao": "Etapa",  
  "tipo": 2,  
  "subProcesso": ""  
}
```

Formulário Processo

O objeto formulário processo é um objeto utilizado nas integrações entre processo e formulário, onde é possível iniciar um item de processo com dados de um formulário preenchidos. O objeto é

composto basicamente pelas propriedades que identificam o processo que será iniciado, o usuário executor, o formulário, e finalmente seus campos e valores.

Nome do campo	Tipo	Descrição
userId	int	Id do Usuário.
processId	int	Id do Processo.
zuriKey	string	Identificador de segurança. Essa propriedade é validada através da chave "Zuri.API.AuthKeys" no arquivo "web.config" da aplicação. Dessa forma, ao informar um "zuriKey" válido é considerado como o usuário logado o usuário que for informado em "userId", ou seja, basta colocar a mesma chave no web.config e no json que a autenticação será considerada válida.
attachmentAlias	string	Apelido do formulário.
paramData	Campo de formulário	Lista de objetos que identificam cada campo de um formulário bem como seus valores preenchidos.

Exemplo do campo zuriKey

Abaixo exemplo da variável configurada no web.config. Ela está localizada na seção <appSettings>.

```
<add key="Zuri.API.AuthKeys" value="B3A7C50A-CF67-4375-B887-B877A216DAD7" />
```

Opções de autenticação

Múltiplas zuriKeys

É possível cadastrar quantas chaves de segurança forem necessárias, assim, por exemplo sistemas diferentes podem ter chaves diferentes, para isso basta separar as chaves por ";" no web.config

```
<add key="Zuri.API.AuthKeys" value="B3A7C50A-CF67-4375-B887-B877A216DAD7; 09812071-1df3-4c78-b374-08941396138e" />
```

Identificação pelo Login ou WinLogon

Ao invés de informar o “userId” (que é numérico), pode ser informado o login ou winlogon do usuário, que a autenticação também ocorre:

```
{
  "login": "admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": []
}

{
  "winLogon": "wf\\admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": []
}
```

Autenticação com login e senha

A forma mais segura de usar esta feature é usando um login e senha de um usuário real, inclusive a senha pode ser passada encriptada (como fica armazenada na tabela de membership) ou de forma aberta (a senha sem encriptação).

A primeira forma, mais segura fica assim:

```
{
  "login": "admin",
  "userPWD": "!@7!ajCj1UT3qPg6sxnDq/KXrw==",
  "zuriKey": "",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": []
}
```

A segunda forma necessita de uma chave no web.config

```
<add key="Zuri.API.UseEncryptedPWD" value="false" />
```

Colocada esta chave, a autenticação poderá ser da seguinte forma:

```
{
  "login": "admin",
  "userPWD": "zuri1234",
  "zuriKey": "",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": []
}
```

Identificação do Processo

Conforme o definido, o processo é identificado pelo seu Id através do parâmetro "processId":

```
{
  "login": "admin",
  "userPWD": "zuri1234",
  "zuriKey": "",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": []
}
```

Mas ele também pode ser chamado através da propriedade "processIdentifier", podendo ser utilizado o GUID do processo ou o nome do mesmo. Mas como o nome não é uma variável única, se houver mais de um processo com o mesmo nome ele dá uma mensagem de erro na identificação do processo.

```
{
  "login": "admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processIdentifier": "Solicitação de Reembolso",
  "attachmentAlias": "PAI",
  "paramData": []
}
```

```
{
  "login": "admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processIdentifier": "3415c2c4-562b-497b-a4f1-d0ac6b1a60c0",
  "attachmentAlias": "PAI",
  "paramData": []
}
```

Campo de formulário

Esse objeto corresponde a um campo do formulário preenchido. Ele possuirá a seguinte estrutura:

Nome do campo	Tipo	Descrição
fieldName	string	Nome do campo.
fieldValue	string	Valor preenchido.
Base64	string	Valor do anexo no formato base64.
listContent	Campo de formulário	Lista de campos filhos

O campo listContent poderá ser preenchido recursivamente quando houver formulários filhos, na seção [Formulário preenchido com campos filhos](#) será exibido exemplo de um objeto preenchido para essa finalidade.

Formulário preenchido

Abaixo um exemplo do objeto Formulário Processo com os campos preenchidos:

```
{
  "login": "admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": [
    {
      "fieldName": "CAMPOX",
      "fieldValue": "aaaa",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "CAMPOY",
      "fieldValue": "1111",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "ARQUIVOTST",
      "fieldValue": "formfiller.png",
      "base64": "iVBORw0KGgoAAAANSUhEUgAAACgAAAAoCMAAAC7IEhfAAAAAB3RJTUUH4gcRDRIydSZOwwAAAA1wSF1zAAALEgAACxIB0t1+/AAAAARnQU1BAACxjwv8YQUAAABgUExURUprc3taxoxrxs695+/v9+/n9+fn97Wc3oRaxrWl3pyE1vf3/8a15970762U1tbG75Rzzv///4Rjxr2t3ufe972l3t7W75x7zqWM1qWE1pR7zoxjxtb078at55Rrzs7G70Q7LfEAAAABdFJOUwBA5thmAAABWU1EQVR42s2V2ZKDIBBFI02zBILiApro+P9/GRwTlwQp32buAyVwqum6fau8XLJTupzkAvkPwdJAno/d9ca/QcH9upHKjb3mNfORijk1yzfLw+KRFbGnwaqhfH1jQxrDauPFDjSSBzlrbf8qwaq7sjcPU05Aahfh/Dw0WatJieO+Iq6gVRUJJ455QohUVXYIWjtddkidq1VfJEA9Vcy8BoDm58PwTY/UiMRkeD0pDpxbTA8tkvITfPto9fvqoZEirbmIgbIaJ60Vlg3v0URAIa/rhoEI8qhFpOJWbJjWivDnwDoG7mLGdKLInmaArfd+oDxuzxqzwlHGkC52HcYse5hw0i1T0ozZwQi/Y5YRww1vywQ4x+zuKGLo8ZEAf2MWHGjbDpQ87PEVM5x9xA8fv20G1ffkdjFLgtuYIUSfnme9jRkozaW8qVgedyJAQ8JZPOEp/SV49vfxBPbyHLiYj+KGAAAAAE1FTkSuQmCC",
      "listContent": []
    },
    {
      "fieldName": "CodigoAutomatico",
      "fieldValue": "3",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "CodigoUsuario",
      "fieldValue": "1",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "GUID",
      "fieldValue": "69c6f12b-4071-48a0-9c52-406805d006ca",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "ParentGUID",
      "fieldValue": "",

```

```

        "base64": "",
        "listContent": []
    }
]
}

```

Formulário preenchido com campos filhos

Abaixo um exemplo de um objeto preenchido com campos filhos:

```

{
  Asdasdas
  {
    "login": "admin",
    "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
    "processId": 6,
    "attachmentAlias": "PAI",
    "paramData": [
      {
        "fieldName": "CAMPOX",
        "fieldValue": "aaaa",
        "base64": "",
        "listContent": []
      }
    ],
    {
      "fieldName": "CAMPOY",
      "fieldValue": "1111",
      "base64": "",
      "listContent": []
    }
    {
      "fieldName": "ARQUIVOTST",
      "fieldValue": "formfiller.png",
      "base64":
        "iVBORw0KGgoAAAANSUhEUgAAACgAAAAoCAMAAC7IEhfAAAAAB3RJTUUH4gcRDRIydSZOwwAAAA1wSF1zAAAEg
        AACxIB0t1+/AAAAARnQU1BAACxjwv8YQUAAABgUExURUprc3taxoxrxs695+/v9+/n9+fn97Wc3oRaxrWl3pyE1
        vf3/8a15970762U1tbG75Rzzv///4Rjxr2t3ufe972l3t7W75x7zqWM1qWE1pR7zoxjxtb078at55Rrzs7G70Q7
        LfEAAAABdFJOUwBA5thmAAABWU1EQVR42s2V2ZKDIBBFi02zBILiApro+P9/GRwTlwQp32buAyVwqum6fau8XLJ
        TupzkAvkPwDjAno/d9ca/QcH9upHKjb3mNfORijk1yzfLw+KRfbGnwaqhFH1jQxrDauPFDjSSBzlrbf8qwaq7sj
        cPU05Aahfh/Dw0WatJie0+Iq6gVRUJJ455QohUVXYIWjtddkidq1VfJEA9Vcy8BoDm58PwTY/UiMRkeD0pDpxbT
        A8tkvITfPto9fvqoZEirbmIgbIaJ60Vlg3v0URAIa/rhoEI8qhFpOJWbJjWivDnwDoG7mLGdKLInmaArfd+oDxu
        zxqzwlHGKc52HcYse5hw0i1T0ozZwQi/Y5YRww1vywQ4x+zuKGLo8ZEAF2MWHGjbdPQ87PEVM5x9xA8fv20G1fF
        kdjFLgtuYIUSfme9jRkozaW8qVgedyJAQ8JZPOEp/SV49vfxBPbyHLiYj+KGAAAAE1FTkSuQmCC",
        "listContent": []
      }
    ],
    {
      "fieldName": "FILHOS",
      "fieldValue": "",
      "base64": "",
      "listContent": [
        {
          "fieldName": "CAMPO1",
          "fieldValue": "222222",
          "base64": "",
          "listContent": []
        }
      ]
    }
  }
}

```

```
, {
  "fieldName": "CATEGORIA",
  "fieldValue": "1",
  "base64": "",
  "listContent": []
}, {
  "fieldName": "ARQUIVOX",
  "fieldValue": "NinkiHistory.png",
  "base64": ""
}
{"ivBORw0KGgoAAAAANSUhEUgAAACgAAAoCAMAAAC7IEhfAAACE1BMVEV+V8F+V8FMaXF+V8F+V8F+V8F+V8F+V8H/////3odzxx7fjp4/V/WMHu6fff8+/6+q+d49vy0ndvr5fbHt+SAWsl08frt6PbOwOiAWcL8/P759/zs5/bJueXo4fTr5fWKZ8fq5PWbfci2FHftOP18vqi2ntvw6/jLvOap6pd6litSLamfy7/mCXMO5pN3v6/fMvOavDcyOa8mNa8jpa4vSMaci7pt728/u5o92Tc8vNvueTcsuSccvdseKwmNmBwsKEXstwyevHtuTRw+n29PuHYsXFs+OJJcbq4/Xov+f v6vfCr+KEX8Tw7PiymtqwdsyRcmrd0++kid0jh90tlNeymr5+pZZe2dgNDEs+PKuuXf1fdKuubd0u6fgtH39PUxEm2afM+8QN+kiNOFYMSObMmZes7IuOWUc8u4o93MveFLu+aBW8l9/P69qd/AreGmjNTk2/Lb007WyuuhhNKUDMy/q+CDXCPEsuOvl9n6+p3l3fp+/v/z8Pmsk9e/rOCpbcnz7/n9/f6qKNbPWOihY8XGteSihtKB8qjiNPazu2Ze8739funjNXhl/CKZsfm3/Pglvc2OnyYec3BruGQbsqxmdny7vnVyOvc0e6NasixmtqGYsxs5vbt5/bxy+yunkteRb8qDXsOWd83QwuI7p97SxerPweijXyzTxure10/Rxonux+rSxonnn4PTYZOy+qudqWukjhbGUAAAAB3RSTlPmqwCA/QFcBUzfGAAaldJRFEUEAGN1QOD40AYGOgu39S2rbvt2zb0tm3btvQTr5PMpcmt8tbpe4ssy+Qiip8UR3ESywViSiYBAUlklBWEJadEwmB68YRBcsPO1++/FP+2MP3EOFY/a7OV82qqGVio0Ad//BozFKoC8LRP69AyORw7mPwsVHOggGWX/x0ATJDml5ZuoR/z0tzBXbtZ6JovilcbC6uX34I9WVZZUVGFztLr1xG/+ZGTQ19+fTIIRFDtv1OKfWNbACAqogESpajh60g89mJLmcL1bAr8ZAUF5Stl6nBrjVk8c0e7lmBVzIUQYnfupHejfajjaamyPfXgdG0Ccwg24DWgyOGDP0ZD2KDVsxa1c6DO97MpR3u4Ms+8uQWyOBi68ED2kin+9Gj0HTPsy0FPLhSpblrnGHrc6zUBny8C921koBTBW4a+bVblk3vEMNAFJiqGXgYuXag+E2UXcz4PGTFhrjzeXQNiJSNvsPNhgTCwFr1LVZDEDKcjOurgl4MO/OI1WrxmK6AP+yK5wIaq8fAg7EG9w3GBmkDNQAp+L0fffsmdVSIFS/Ej4Y7H441N8EmY40InS02PwfVhcZdXkwNwtQDJ3HEe48lo+fJyq7r9xJTas4MwxWF3pt2Y1D7bT27FMc57MqjeM2nx5PoUKnr3lQVC/hlrh2Up2bLbz1Gn82akwmHjwxX16iiysG2c3UjCLCZW59L62lpMt2cfuooSnkwsnz5ojCsVU40mg04Unxkx0oRFnPRCW5Cqv0RNNeA5nNs dSEpJF6eVICIVHL6SCL82CL/ZCbX9/AWSLiJoebFAeAAAAAE1FTkSuQmCC",
  "listContent": []
}, {
  "fieldName": "NETOS",
  "fieldValue": "",
  "base64": "",
  "listContent": [
    [
      {
        "fieldName": "CAMPOLL",
        "fieldValue": "LLVALOR1",
        "base64": "",
        "listContent": []
      }
    ],
    [
      {
        "fieldName": "CAMPOLL",
        "fieldValue": "LLVALOR2",
        "base64": "",
        "listContent": []
      }
    ]
  ]
}
```


itemId	int	Id do item de processo criado.
--------	-----	--------------------------------

Exemplos de respostas

Abaixo exemplos de respostas:

Item criado com sucesso

```
{
  "StatusCode": 200,
  "Message": "Item created successfully!",
  "ItemId": 19148
}
```

Erro de autenticação

```
{
  "StatusCode": 403,
  "Message": "Authentication failed for user!",
  "ItemId": 0
}
```

Erro de definição de processo

```
{
  "StatusCode": 404,
  "Message": "Process not found!",
  "ItemId": 0
}
```

Erro no apelido do formulário

```
{
  "StatusCode": 400,
  "Message": "Attachment provider cannot be created!",
  "ItemId": 0
}
```

Erro de processamento

A resposta abaixo corresponde a qualquer erro de processamento, retornando o StatusCode 500.

```
{
  "StatusCode": 500,
  "Message": "Error on request processing!",
  "ItemId": 0
}
```

Tabelas de Tipos

HTTP Status Code

A tabela abaixo lista os códigos de resposta HTTP que a API poderá retornar.

200	OK – tudo ocorreu conforme esperado
201	(Created) Criado

400	(Bad Request) Requisição inválida
401	(Unauthorized) Não autorizado (falta informação de autenticação, usuário/senha errado ou token de autenticação inválido/vencido)
403	(Forbidden) Proibido (acesso negado para o usuário)
404	(Not Found) Não encontrado
500	(Internal Error) Erro interno no servidor
501	(Not Implemented) Significa que o processamento tem componentes que só podem rodar no browser, não pode ser executado na APP

Tipos de Menu

A tabela abaixo lista os tipos de menu existentes na api mobile.

Processes	Menu para tela de processos. Neste tipo de menu a propriedade URL contém a url do endpoint que deve ser chamado para retornar a lista de processos para a aplicação mobile
Inbox	Menu para tela de itens de processo. Neste tipo de menu a propriedade URL contém a url do endpoint que deve ser chamado para listar os itens a serem exibidos
WebView	Menu para tela de webview. Neste tipo de menu a propriedade URL contém a url do webview a ser chamada
Map	Menu para tela de mapa. Neste tipo de menu a propriedade URL contém a url do endpoint que deve ser chamado para listar os pontos do mapa solicitado.
OfflineForms	Menu para tela de formulários offline. Neste tipo de menu a propriedade URL contém a url do endpoint que deve ser chamado para retornar as configurações dos formulários que podem ser preenchidos offline
StartItem	Menu para tela de início de item de processo. Neste tipo de menu a propriedade URL contém a url do endpoint que deve ser chamado para iniciar um item do processo configurado neste menu.

Tipos de objeto Theme

A tabela abaixo lista os tipos de objeto theme existentes

Color	Indica que a informação existente no campo Value do objeto é uma cor.
Image	Indica que a informação existente no campo Value do objeto é uma url de imagem.

Tipos de Execução

A tabela abaixo lista os tipos de execução existentes na api REST.

b4aa1704-c8b5-46ae-8e70-ce00253ee023	Decisão
1b34d0fa-d951-425a-97f7-7f1c6652794d	Formulário

Estado de Execução

A tabela abaixo lista os estados de execução existentes na api REST.

0	Pending. Requer ação do usuário (tomar decisão, preencher formulário...). Exibir tela de acordo com o execType
1	Finished. Execução de etapa finalizada. Fechar a tela de detalhe e voltar para a listagem da caixa de entrada.
2	Error. Ocorreu um erro durante a execução desta etapa do ítem. Se houver uma mensagem em execMessage, ela deve ser exibida.
3	Cancelled. Execução cancelada. Se houver uma mensagem em execMessage, ela deve ser exibida.

Tipo de Campo

A tabela abaixo lista os tipos de campo de formulário disponíveis.

text	Campo de tipo texto
Int	Campo de tipo número inteiro.
decimal	Campo de tipo número decimal.
boolean	Campo de tipo boolean (true, false – checkbox)
scale	Campo slider de valor decimal variando de 0 a 1 (termômetro ou slider)
rating	Campo de classificação 5 estrelas, variando de 0 a 1 (cada estrela vale 0.2)
date	Campo de tipo data
datetime	Campo de tipo data e hora
time	Campo do tipo hora
datetimetype	Campo de tipo data e hora, com segundo campo de hora indicando hora final (por exemplo, para agendamento de reuniões)
label	Campo contendo uma mensagem para ser exibida no formulário
position	Campo de geocoordenada
map	Campo para exibir um ponto em mapa na tela
option	Campo dropdown para seleção de uma opção
multiple	Campo dropdown para múltipla seleção
action	Botão de ação
group	Grupo de campos

API

A seguir estão todos os novos endpoints expostos pelo Zuri para aplicações mobile e de terceiros trabalharem com itens de workflow.

Valores **entre** colchetes [] nas definições de url dos endpoints abaixo são dados opcionais.

Valores **seguidos** de colchetes [] nas definições de tipos de dados ou retorno indicam que é um array do tipo informado.

Informações entre chaves {} nas definições de url dos endpoints indicam parâmetros que devem ser substituídos pelos valores desejados, conforme descrito em cada caso.

Alguns endpoints suportam método PUT ou POST. Se tentar com PUT e não funcionar, tente novamente usando POST. Isso acontece por uma limitação na configuração de alguns servidores que nos impede de utilizar os verbos PUT e DELETE sem interferir na configuração de outros sites existentes no mesmo servidor.

Os request PUT ou POST que esperam um objeto json no request devem ter o content-type do request definido corretamente para funcionar. Mesmo se o conteúdo e url estiverem certos, o request vai falhar se o content-type do request não estiver definido para “application/json”. Requests do tipo GET não precisam disso.

Identificação do device

A aplicação vai suportar notificações push em uma versão futura. Para que isso funcione é necessário que os aplicativos mobile se identifiquem. Se o aplicativo quiser se cadastrar para receber push notifications, basta que inclua uma string no formato abaixo no header “User-Agent” do request:

Zuri/{versao} ({plataforma}:{deviceId});{nome}

Nome do campo	Descrição	Exemplo
versao	Número de versão do aplicativo mobile que está acessando a API	1.0
plataforma	Identificador da plataforma do aplicativo (iOS, Android, UWP)	iOS
deviceId	String do id do aparelho para ser usado em notificações push.	Cada plataforma tem um formato

nome	Nome do aparelho. Esta informação é opcional, mas é recomendado que os aplicativos mobile informem para que o usuário possa identificar e remover aparelho da lista de notificações push usando o Zuri.	My iPhone
------	---	-----------

É recomendável que haja uma tela no aplicativo mobile onde o usuário possa escolher se quer ou não receber notificações push do Zuri. Se quiser, basta chamar a API usando um User-Agent no formato descrito acima. Se quiser remover o aparelho das notificações push basta acrescentar um sinal de menos imediatamente antes da palavra Zuri no User-Agent. O User-Agent pode conter outras informações além dessas (pode ser acrescentado ao User-Agent padrão ao invés de substituir) que o Zuri irá entender corretamente.

Notificações push serão suportadas apenas nos aplicativos mobile oficiais do Zuri.

Exemplo de header User-Agent se identificando para registrar um aparelho iOS de deviceId = AAAAAAAAAAAAAA para push notification.

```
User-Agent: Zuri/2.0 (iOS:AAAAAAAAAAAA;iPhone)
```

Autenticação

GET /api/auth/

Requer Autenticação: Sim (Basic)

Descrição: Autenticação da API.

Retorno sucesso

HttpStatus: 200

Corpo da resposta: String contendo o token de autenticação “Bearer” a ser usado nos outros endpoints da API

Retorno erro

HttpStatus: 401 Unauthorized

Descrição: Ignorar o corpo da resposta se for diferente de 200. Se a chamada a este endpoint devolver 401 significa que o usuário e senha informados estão incorretos ou o usuário está desativado/bloqueado. Se qualquer outro endpoint da API retornar 401 significa que o token “Bearer” está incorreto ou expirado e deve ser solicitado um novo usando /api/auth/.

A maioria dos endpoints listados neste documento requerem um token de autenticação para funcionar. O processo de autenticação funciona de forma semelhante ao modelo de autenticação OAuth, só que mais simplificado. O aplicativo deve realizar um GET na url /api/auth/ passando o usuário e senha no HTTPHeader “Authorization” do request seguindo o padrão Basic. O padrão basic consiste na palavra “Basic “ + uma string em formato Base64 composta do usuário e senha separados por “:”. Quando autenticado com sucesso esse endpoint retorna uma string que é o token de autenticação que deve ser utilizado na próximas chamadas à API.

Abaixo há um fragmento de código em VB.NET mostrando como configurar um web request com autenticação em formato Basic.

```
Dim c = New WebClient()  
Dim authRawString As String = String.Concat("username", ":", "password")  
Dim authUtf8Bytes As Byte() = Encoding.UTF8.GetBytes(authRawString)  
Dim authBase64Encoded As String = Convert.ToBase64String(authUtf8Bytes)  
c.Headers(HttpRequestHeader.Authorization) = String.Concat("Basic ", authBase64Encoded)  
...
```

O header de http gerado pelo exemplo acima será:

```
Authorization: Basic dXN1cm5hbWU6cGFzc3dvcmQ=
```

A autenticação em formato basic só deve ser utilizada neste endpoint. Os outros endpoints da API devem fazer autenticação em formato Bearer (Token), informando o token recebido em /api/auth/. Abaixo há um exemplo de header de autenticação em formato Bearer. O texto em vermelho no exemplo é o token que foi retornado pela api de autenticação.

```
Authorization: Bearer  
!@7!w/KEspvv1r3m3gSA+BZnvnEv4IsMXPPokjRTY7WglHi14HNFhpEl18CSS5WXmov=
```

Configurações Mobile

GET /api/zuri/settings/
Requer Autenticação: Não
Descrição: Obtém as configurações do aplicativo mobile (menus, cores, imagens de background...)

Retorno
HttpStatus: 200
Corpo da resposta: [Zuri Settings](#)

Pontos no Mapa

GET /api/zuri/map/{nomeMapa}/{coord}/
Requer Autenticação: Sim (Bearer)
Descrição: Obtém uma coleção de pontos a serem exibidos em um mapa.

A url para obter os pontos no mapa é obtida em um objeto do tipo Menu retornado pela api de configurações. A url retornada terá o valor {coord} que deverá ser substituído conforme descrito abaixo. O {nomeMapa} já será fornecido na url retornada nas configurações.

Parâmetro	Tipo	Descrição
-----------	------	-----------

coord	Double,double	O parâmetro {coord} na url deve ser uma string contendo a latitude e longitude da coordenada separados por “,” e usando ponto como casa decimal. A latitude deve ser um valor entre -90 e 90 e a longitude entre -180 e 180. É importante que seja adicionada a barra no final da url, após as coordenadas, para que o request seja interpretado corretamente. Exemplo: GET /api/zuri/map/default/-23.564620,-46.690923/
-------	---------------	--

Retorno

HttpStatus: 200

Corpo da resposta: [Ponto no Mapa](#)[] (array de pontos no mapa)

Lista de Processos

GET /api/zuri/procs/

Requer Autenticação: Sim (Bearer)

Descrição: Lista todos os processos que o usuário pode iniciar ou onde haja itens na caixa de entrada do usuário.

Retorno

HttpStatus: 200

Corpo da resposta: [Processo](#)[] (array de processos)

Lista de itens na Caixa de Entrada

GET /api/zuri/items/inbox[?procid={procid}]

Requer Autenticação: Sim (Bearer)

Descrição: Lista itens na caixa de entrada do usuário. A url usada para listar os itens de apenas um processo é retornada como parte do objeto Processo da Lista de processos.

Parâmetro	Tipo	Descrição
procid	Integer	Opcional. Id do processo para filtrar a lista de itens.

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem na caixa de entrada](#)[] (array de itens)

Lista de itens em “Meus itens”

GET /api/zuri/items/mine[?procid={procid}]

Requer Autenticação: Sim (Bearer)

Descrição: Lista itens que o usuário criou ou já participou. A url usada para listar os itens de apenas um processo é retornada na lista de links no objeto Processo.

Parâmetro	Tipo	Descrição
procid	Integer	Opcional. Id do processo para filtrar a lista de itens.

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem na caixa de entrada](#)[] (array de itens)

Lista de itens em Acompanhamento

GET /api/zuri/items/following[?procid={procid}]

Requer Autenticação: Sim (Bearer)

Descrição: Lista itens para acompanhamento do usuário. A url usada para listar os itens de apenas um processo é retornada na lista de links no objeto Processo.

Parâmetro	Tipo	Descrição
procid	Integer	Opcional. Id do processo para filtrar a lista de itens.

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem na caixa de entrada](#)[] (array de itens)

Detalhe de ítem na caixa de entrada

GET /api/zuri/items/{id}/{step}/

Requer Autenticação: Sim (Bearer)

Descrição: Obtém os detalhes de um ítem na caixa de entrada, meus itens ou acompanhamento. A url exata a ser chamada está na propriedade detailsUrl dos objetos do tipo [Ítem na caixa de entrada](#) retornados na listagem da caixa de entrada. Se o item não estiver na caixa de entrada do usuário, mas estiver na lista de “meus itens” ou acompanhamento o usuário poderá ver o item mas não terá permissão para realizar alterações em formulários ou tomar decisões (a lista de decisões nem será retornada pela API).

Parâmetro	Tipo	Descrição
id	Integer	Id do ítem processo

step	Integer	Id da etapa item
------	---------	------------------

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem](#)

Executar Ítem

POST ou PUT /api/zuri/items/{id}/{step}/exec/

Requer Autenticação: Sim (Bearer)

Descrição: Executa um ítem do Zuri. A url exata de execução de um ítem está na propriedade “execUrl” do objeto [Ítem](#).

Content-Type: application/json

Corpo do Request: [Ítem](#)

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem](#)

Iniciar Processo

POST /api/zuri/procs/{id}/items/

Requer Autenticação: Sim (Bearer)

Descrição: Inicia um novo ítem do processo informado. A url exata para iniciar ítems de processo está na propriedade startUrl do objeto [Processo](#).

Corpo do Request: Vazio

Content-Type: não precisa

Retorno

HttpStatus: 200

Corpo da resposta: [Ítem](#)

Listar formulários que podem ser preenchidos offline

GET /api/ninki/forms/offline/

Requer Autenticação: Sim (Bearer)

Descrição: Devolve as configurações de todos os formulários que o usuário pode preencher offline.

Retorno

HttpStatus: 200

Corpo da resposta: [Definição de Formulário](#)[] (array de definições de formulário)

Atualizar dados de formulário já preenchido (checkout)

POST ou PUT /api/ninki/forms/{formid}/items/{id}/

Requer Autenticação: Sim (Bearer)

Descrição:

Corpo do request: [Dados de Formulário](#) (dados do formulário a atualizar)

Content-Type: application/json

Retorno
HttpStatus: 200

Atualização de sources de formulário

POST ou **PUT** /api/ninki/forms/{formid}/items/{id}/sources
Requer Autenticação: Sim (Bearer)
Descrição: Atualiza os dados de um formulário que tenha campos com dependencia.
Corpo do request: [Dados de Formulário](#) (dados do formulário a atualizar, o campo trigger do objeto deve conter o name do campo que causou a atualização de sources)
Content-Type: application/json

Retorno
HttpStatus: 200
Corpo da resposta: [Sources de Formulário](#) (lista de sources atualizados do formulário)

Detalhes de histórico de um item de processo

GET /api/zuri/items/{itemWorkflowId}/details/
Requer Autenticação: Sim (Bearer)
Descrição: Obtém o histórico do item de processo ordenado por ordem cronológica.
Content-Type: application/json

Retorno
HttpStatus: 200
Corpo da resposta: [Item Detalhado](#)

Iniciar Processo com dados de formulário

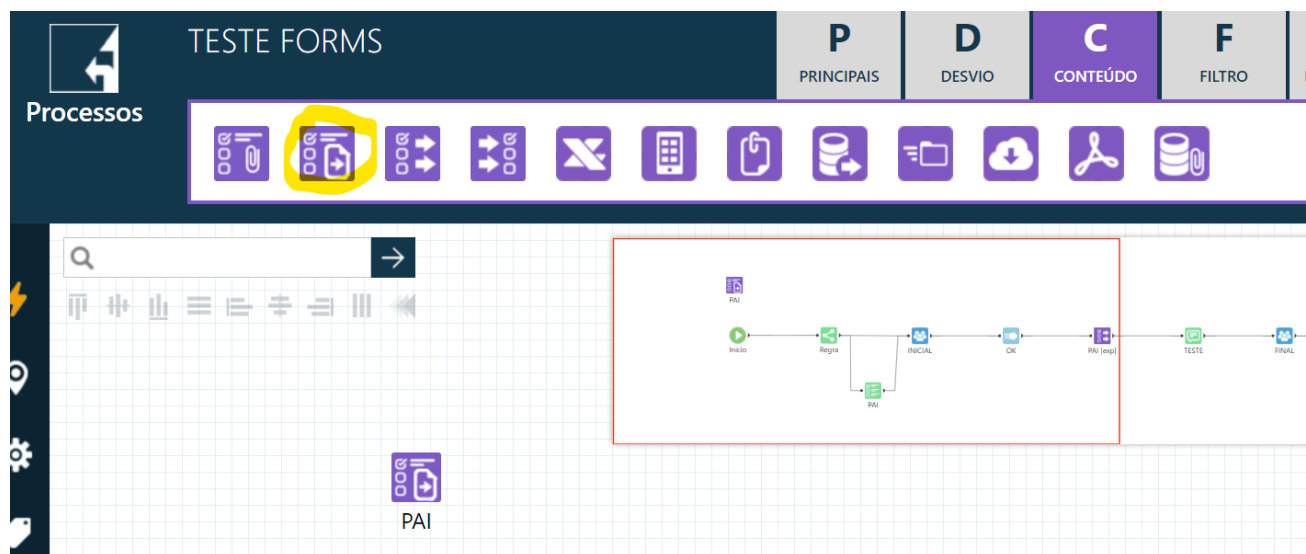
POST /zuriservices/startitem
Requer Autenticação: Não
Descrição: Inicia um novo item do processo informado e preenche dados de um formulário.
Corpo do Request: [Formulário Processo](#)
Content-Type: application/json

Retorno
HttpStatus: 200
Corpo da resposta: [Item Processo Formulario](#)

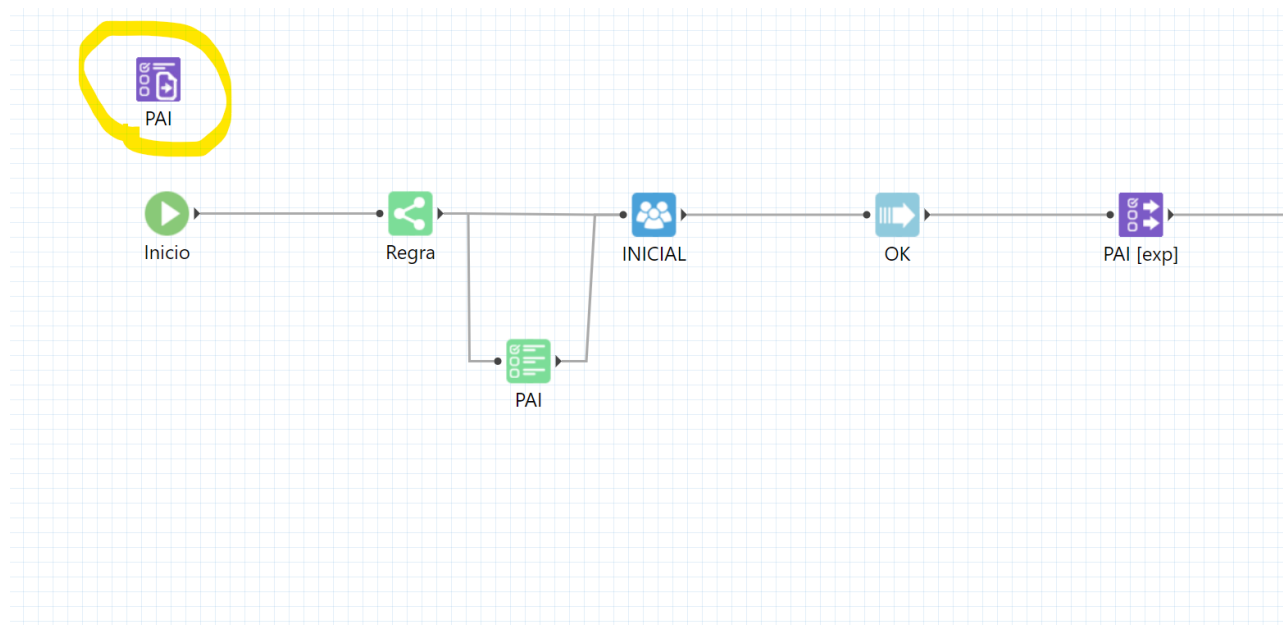
Segue abaixo a configuração do formulário no desenho do processo na ferramenta:

Definição do formulário no fluxo

O novo módulo para definição do formulário foi colocado na aba de conteúdo:



O formulário pode ser declarado no fluxo usando este módulo, mas este pode ser colocado livremente no fluxo, pois só declara a existência do formulário no mesmo, ele não tem conexão nenhuma:



Ele não tem conectores e não é obrigatório

Se já existir um formulário no processo ele não precisa ser colocado.

Segue abaixo a configuração do módulo no desenho do processo:

zuri9.produtos.wf/WFAdmin/WFWorkflow/modules/SelectForm.aspx?ITEMID=KSD...

Não seguro | zuri9.produtos.wf/WFAdmin/WFWorkflow/modules/SelectForm.aspx?ITE...

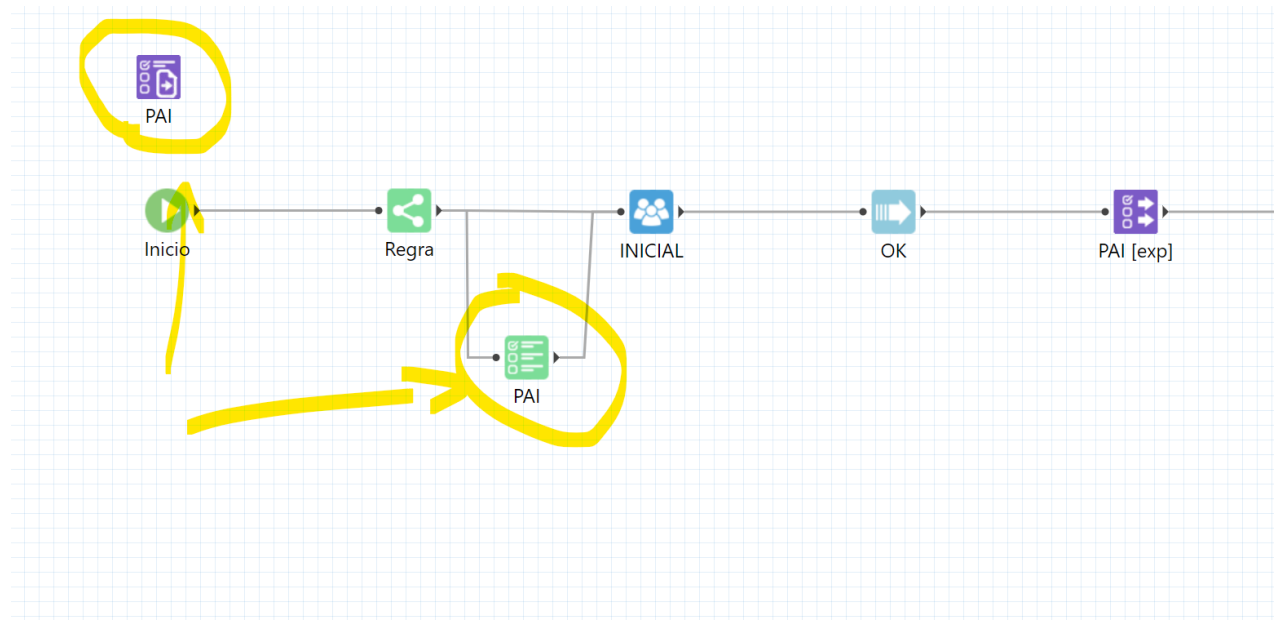
Define formulário

Formulário

PAI

ok cancelar

Geração dos Anexos



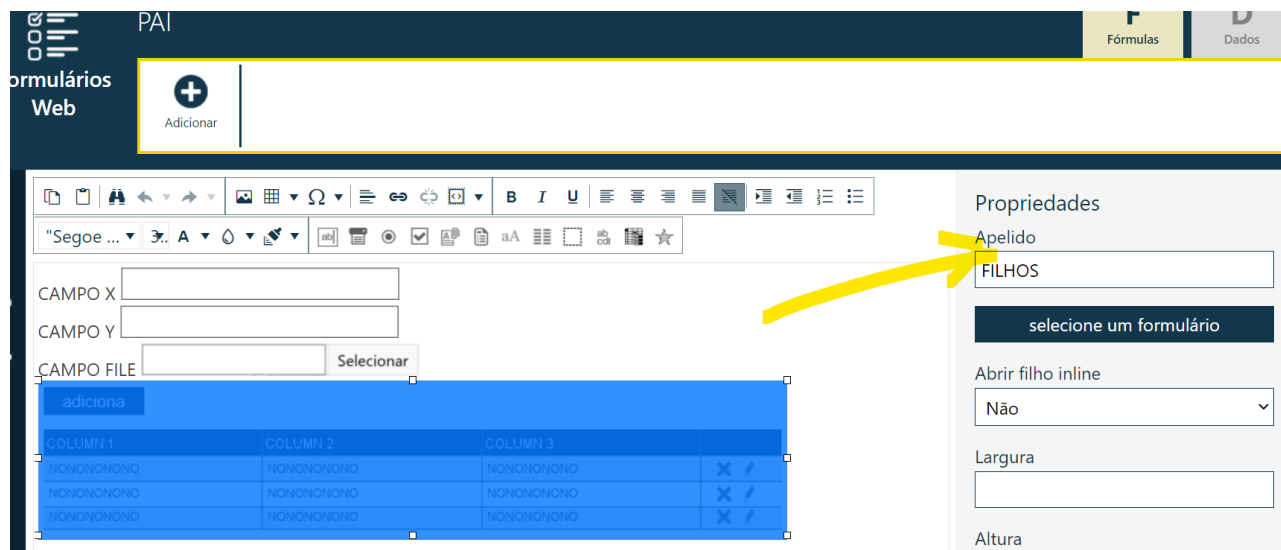
A geração dos anexos nos dois casos acima é realizada de forma idêntica, inclusive com a montagem dos arquivos como anexos.

Só são gerados anexos para pais e filhos, pois netos ficam sem referência. Se for necessário trazer informações dos netos para dentro do fluxo tem que ser utilizado o módulo Anexa Subformulário em um loop (da mesma forma que se faz quando tem um formulário com filhos e netos)

Os dados de netos, bisnetos e qualquer outro nível fornecido na chamada são devidamente gravados no banco do Forms, somente os anexos não são gerados.

ATENÇÃO, não é o nome do form filho e sim o nome do filho no pai

Configuração do form:



Exemplo do json enviado no corpo da requisição:

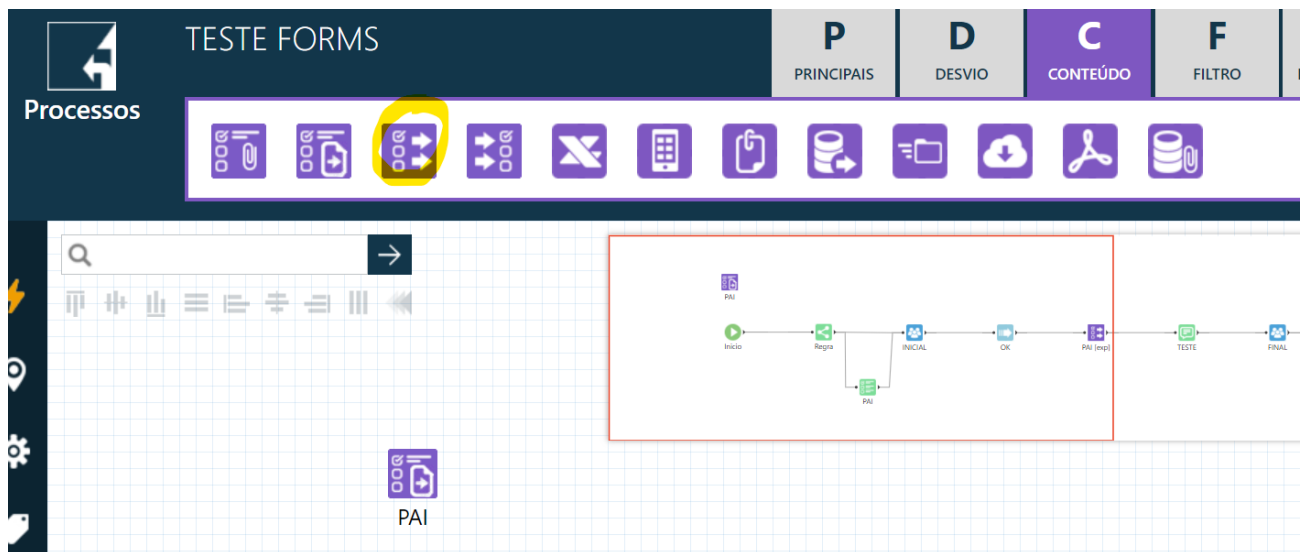
```
{
  "fieldName": "FILHOS",
  "fieldValue": "",
  "base64": "",
  "listContent": [
    {
      "fieldName": "CAMP01",
      "fieldValue": "222222",
      "base64": "",
      "listContent": []
    }
  ],
  "fieldName": "CATEGORIA",
  "fieldValue": "1",
}
```

Módulo de exportação de Dados

O módulo de exportação está junto com o módulo de seleção de formulário. Esse novo módulo é responsável pela exportação de dados de um formulário para o formato json.

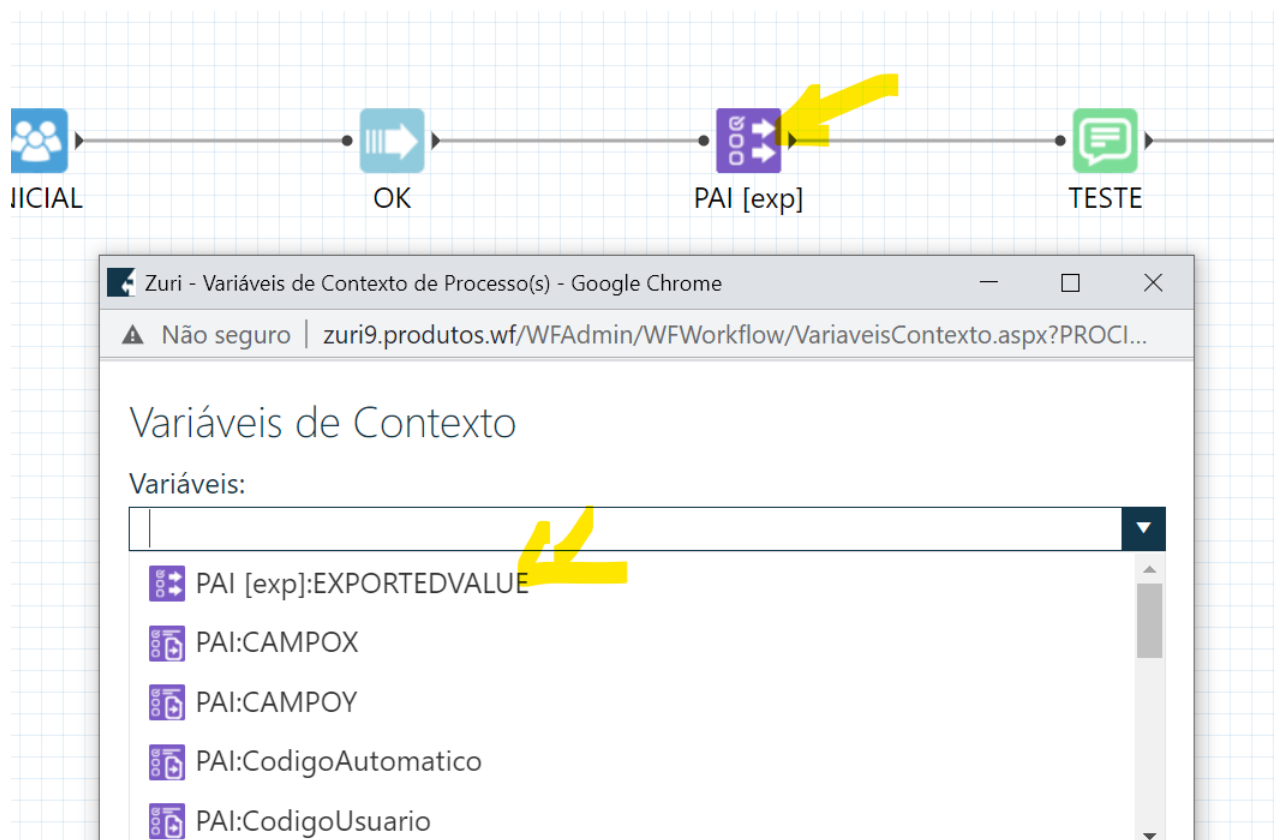
O json exportado do formulário segue a mesma estrutura utilizada na requisição da api [Início de Processo com dados](#).

Abaixo a identificação do módulo:



Em sua configuração, é necessário somente a seleção do formulário que deseja exportar os dados.

O módulo faz a exportação de todos os níveis (pais, filhos, netos, bisnetos,...) em formato json para uma variável de contexto:



Estrutura do json:

```
1  {
2    "processId": "10",
3    "processName": "TESTE FORMS",
4    "itemWorkflowId": "89",
5    "attachmentAlias": "PAI",
6    "paramData": [
7      {
8        "fieldName": "CAMPOX",
9        "fieldValue": "aaaa",
10       "base64": "",
11       "listContent": []
12     },
13     {
14       "fieldName": "CAMPOY",
15       "fieldValue": "1111",
16       "base64": "",
17       "listContent": []
18     }
19   ]
20 }
```

Exemplo de utilização

Abaixo segue um exemplo simples de utilização da API mostrando o conteúdo dos requests e respostas, para que fique um pouco mais claro como a api deve ser utilizada. Alguns headers dos requests e respostas abaixo foram removidos por não serem relevantes para o entendimento do funcionamento da API.

1 – A aplicação obtém as configurações. Este request não requer autenticação e pode ser usado para validar se a url do Zuri informada pelo usuário está correta.

Request

```
GET http://zuri/api/zuri/settings/ HTTP/1.1
```

Resposta

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json; charset=utf-8
```

```
{
  "theme": [
    { "name": "Background", "type": "Color", "value": "#FF0000" },
    { "name": "Menu", "type": "Color", "value": "#FF0000" }
  ],
  "menus": [
    { "name": "Processos", "type": "Processes", "url": "/api/zuri/procs/" },
    { "name": "Inbox Completa", "type": "Inbox", "url": "/api/zuri/items/inbox/" },
    { "name": "Pontos de Interesse", "type": "Map", "url": "/api/zuri/map/pontosdeinteresse/{coord}/" },
    { "name": "Google", "type": "WebView", "url": "http://google.com/" },
    { "name": "Formulários Offline", "type": "OfflineForms", "url": "/api/ninki/forms/offline/" },
    { "name": "Iniciar Visita", "type": "StartItem", "url": "/api/zuri/procs/19/items/" }
  ]
}
```

O aplicativo atualiza as cores da e imagens de fundo de acordo com o que for retornado pela api de settings e monta o menu interno.

2 – A aplicação chama a API de autenticação em modo basic para o usuário “usuario”, cuja senha é “usuario”. A aplicação também está se registrando para push notification.

Request

```
GET http://zuri/api/auth/ HTTP/1.1
```

```
Authorization: Basic dXN1YXJpbzplc3Vhcm1v
```

```
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)
```

Resposta

```
HTTP/1.1 200 OK
Content-Type: text/plain; charset=utf-8
```

```
!@7!IHLiBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
```

Guarda o token recebido para usar no header de autenticação dos próximos requests.

O usuário clica na opção “Processos” do menu

3 – A aplicação obtém a lista de processos

Request

```
GET http://zuri/api/zuri/procs/ HTTP/1.1
Authorization: Bearer
!@7!IHLiBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)
```

Resposta

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

[{"name":"teste","inboxCount":0,"startUrl":"/api/zuri/procs/2/items/","inboxUrl":"/api/zuri/items/inbox/?procid=2"}, {"name":"teste 2","inboxCount":0,"startUrl":"/api/zuri/procs/3/items/","inboxUrl":"/api/zuri/items/inbox/?procid=3"}]
```

O usuário clica no processo “teste 2”

4 – O usuário seleciona o processo “teste 2” e a aplicação obtém a lista de itens da caixa de entrada do usuário usando a url da propriedade inboxUrl.

Request

```
GET http://zuri/api/zuri/items/inbox/?procid=3 HTTP/1.1
Authorization: Bearer
!@7!IHLiBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)
```

Resposta


```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
```

```
[{"detailsUrl":"/api/zuri/items/1/2/","name":"teste 2_1","id":1,"status":"Inicializado","step":"Etapa 1","info":""}, {"detailsUrl":"/api/zuri/items/2/6/","name":"teste 2_2","id":2,"status":"Inicializado","step":"Etapa 1","info":""}, {"detailsUrl":"/api/zuri/items/6/12/","name":"teste 2_6","id":6,"status":"Inicializado","step":"Etapa 1","info":""}]
```

5 – O usuário seleciona o item “teste 2_1” e a aplicação obtém os detalhes desse item usando a url que está na propriedade detailsUrl do objeto.

Request

```
GET http://zuri/api/zuri/items/1/2/ HTTP/1.1
Authorization: Bearer
!@7!IHLiBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)
```

Resposta

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"id":1,"activityId":0,"name":"teste 2_1","status":"Inicializado","execUrl":"/api/zuri/items/2/6/exec/","execMessage":null,"execType":"b4aa1704-c8b5-46ae-8e70-ce00253ee023","execStatus":0,"execData":{"message":"Aprovar o item de teste?","decisions":[{"id":4,"name":"Aprovar"}, {"id":5,"name":"Reprovar"}]}}
```

6 – A aplicação vê que o tipo de execução retornado é “tomada de decisão” (execType) e mostra para o usuário uma tela de tomada de decisão com a mensagem “Aprovar o item de teste?” e os botões de decisão “Aprovar” e “Reprovar”. O usuário seleciona “Reprovar” e a aplicação envia os dados da decisão tomada para a API usando a url que está na propriedade execUrl. O objeto enviado para o servidor é o mesmo que foi recebido anteriormente, mudando apenas o valor de execData, que agora deve conter apenas a decisão que foi tomada.

Request

```
POST ou PUT http://zuri/api/zuri/items/1/2/exec/ HTTP/1.1
Authorization: Bearer
!@7!IHliBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
Content-Type: application/json; charset=utf-8
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)

{"id":1,"activityId":0,"name":"teste
2_1","status":"Inicializado","execUrl":"/api/zuri/items/2/6/exec/","execM
essage":null,"execType":"b4aa1704-c8b5-46ae-8e70-
ce00253ee023","execStatus":0,"execData":{"id":5,"name":"Reprovar"}}
```

Resposta

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8

{"id":1,"activityId":0,"name":"teste
2_1","status":"Reprovado","execUrl":"/api/zuri/items/2/6/exec/","execMess
age":null,"execType":"","execStatus":1,"execData":null}
```

7 – A aplicação identifica que o estado da execução é do tipo “Finished” e retorna para a tela de listagem da caixa de entrada do processo, repetindo o request feito no passo 4.

Se durante o processamento, algum dos requests retornar HTTP status 401, o passo 2 deverá ser executado novamente para obter um novo token Bearer para utilizar nos próximos requests.

8 – Obter detalhes de um item com seu histórico ordenado por ordem cronológica.

Request

```
GET http://zuri/api/zuri/items/1/details HTTP/1.1
Authorization: Bearer
!@7!IHliBunXzuIuWHjKAiggsefpZxfnC+6l97gbSZPyJ7pZt1SzbFbJcgejrAwRZ8sL
User-Agent: Zuri/1.2 (iOS:123456; My iPhone 6)
```

Resposta

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

```
{
  "id": 25,
  "nome": "Item de processo de teste",
  "status": "Inicializado",
  "dataInicio": "2021-03-05T19:59:47",
  "dataFim": null,
  "readOnly": false,
  "usuOwner": 1,
  "areaOrigem": 1,
  "itemOrigem": 0,
  "processoOrigem": 6,
  "param1": "",
  "param2": "",
  "param3": "",
  "param4": null,
  "param5": null,
  "historico": [
    {
      "executor": "Administrador",
      "execucao": "05/03/2021 19:59",
      "etapa": "Inicio",
      "descricao": "Iniciou a etapa Inicio",
      "status": "Inicializado",
      "acao": "Etapa",
      "tipo": 2,
      "subProcesso": ""
    },
    {
      "executor": "Administrador",
      "execucao": "05/03/2021 20:00",
      "etapa": "Inicio",
      "descricao": "Formulário grid - formulario pai
preenchido",
      "status": "Inicializado",
      "acao": "Formulario Web",
      "tipo": 3,
      "subProcesso": ""
    },
    {
      "executor": "Administrador",
      "execucao": "05/03/2021 20:00",
      "etapa": "Inicio",
      "descricao": "Finalizou a etapa Inicio",
```

```

        "status": "Inicializado",
        "acao": "N/A",
        "tipo": 22,
        "subProcesso": ""
    },
    {
        "executor": "Administrador",
        "execucao": "05/03/2021 20:00",
        "etapa": "Etapas de Início",
        "descricao": "Iniciou a etapa Etapas de Início",
        "status": "Inicializado",
        "acao": "Etapa",
        "tipo": 2,
        "subProcesso": ""
    }
]
}

```

9 – Iniciar um item de processo com dados de formulário preenchidos.

Request

GET http://zuri/ZuriServices/StartItem HTTP/1.1

```

{
  "login": "admin",
  "zuriKey": "B3A7C50A-CF67-4375-B887-B877A216DAD7",
  "processId": 6,
  "attachmentAlias": "PAI",
  "paramData": [
    {
      "fieldName": "CAMPOX",
      "fieldValue": "aaaa",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "CAMPOY",
      "fieldValue": "1111",
      "base64": "",
      "listContent": []
    },
    {
      "fieldName": "ARQUIVOTST",
      "fieldValue": "formfiller.png",

```

```
"base64":  
"iVBORw0KGgoAAAANSUhEUgAAACgAAAAoCMAAAAC7IEhfAAAB3RJTUUH4gcRDRIydSZ0wwAA  
AA1wSF1zAAAEgAACxIB0t1+/AAAAARnQU1BAACxjwv8YQUAAABgUEXURUprc3taxoxrs695  
+/v9+/n9+fn97Wc3oRaxrWl3pyE1vf3/8a15970762U1tbG75Rzzv///4Rjxr2t3ufe972l3t  
7W75x7zqWM1qWE1pR7zoxjxtb078at55Rrzs7G70Q7LfEAAAABdFJOUwBA5thmAAABWU1EQVR  
42s2V2ZKDIBBFi02zBILiApro+P9/GRwTlwQp32buAyVwqum6fau8XLJTupzkAvkPwDJAno/d  
9ca/QcH9upHKjb3mNfORijk1yzfLw+KRFbGnwaqhFh1jQxrDauPFDjSSBzlrbf8qwaq7sjcPU  
05Aahfh/DwOWatJie0+Iq6gVRUJJ455QohUVXYIWjtddkidq1VfJEA9Vcy8BoDm58PwTY/UiM  
RkeDOPdpXbTA8tkvITfPto9fvqoZEirbmIgbIaJ60Vlg3v0URAIa/rhoEI8qhFpOJWbJjWivD  
nwDoG7mLGdKLiNmaArfd+oDxuzxqzw1HGkC52HcYse5hw0i1TOozZwQi/Y5YRww1vywQ4x+zu  
KGLO8ZEAf2MWHGjbDpQ87PEVM5x9xA8fv20G1fFkdjFLgtuYIUSfnme9jRkozaW8qVgedyJAQ  
8JZPOEp/SV49vfxBPbyHLiYj+KGAAAAAE1FTkSuQmCC",  
  "listContent": []  
}  
, {  
  "fieldName": "FILHOS",  
  "fieldValue": "",  
  "base64": "",  
  "listContent": [  
    {  
      "fieldName": "CAMPO1",  
      "fieldValue": "222222",  
      "base64": "",  
      "listContent": []  
    }  
  ],  
  {  
    "fieldName": "CATEGORIA",  
    "fieldValue": "1",  
    "base64": "",  
    "listContent": []  
  }  
  , {  
    "fieldName": "ARQUIVOX",  
    "fieldValue": "NinkiHistory.png",  
    "base64":  
"iVBORw0KGgoAAAANSUhEUgAAACgAAAAoCMAAAAC7IEhfAAACE1BMVEEV+V8F+V8FMAxF+V8F+  
V8F+V8F+V8F+V8H///+3odz7fj4/V/WMHu6ff8+/6+q+D49vy0ndvr5fbHt+SAwSL08frt6  
PbOwOiAWcL8/P759/zs5/bJueXo4fTr5fWKZ8fq5PWbfc/i2fHFtOP18vq1ntvw6/jLv0a6pd  
6litSLaMfy7/mCXM05pN3v6/fMv0aVdcy0a8mNa8jp4vSMaci7pt728/u5o92Tc8vNvueTcsu  
SccvDseKwmNmBWsKEXsTWyevHtuTRw+n29PuHYsXFs+OJZcbq4/XOv+fv6vfCr+KEX8Tw7Piy  
mtqWdsyRcMrd0++kid0jh90tlNeym9r5+PzZze2dgNDEs+PKuuXf1fDKuubd0u6fgtH39PuXe  
M2afM+8qN+kiNOFYMS0bMmZes7Iu0WUc8u4o93MvefLu+aBW8L9/P69qd/AreGmjNTk2/Lb00  
7WyuuhhNKUdMy/q+CDXCPEsu0v19n6+P3l3fP+/v/z8Pmsk9e/rOCPbcnz7/n9/f6qkNbPwOi  
HY8XGteSihtKQb8qjiNPazu2Ze8739funjNXh1/CKZs-fm3/Pg1vC2oNyYec3BruGQbsqxmndy  
7vnVy0vc0e6NasixmtqGYsXs5vbt5/bXy+yRkteRb8qDXs0Wd83Qwui7p97SxerPwejXyuzTx  
ure10/RxOnUx+rSxOnn4PTYzOy+quDQwukjhbGUAAAAB3RSTlPmqwCA/QFcbUzifgAAAldJRE  
FUeAGN1QOD40AYgOGU39S2rbVt2zb0tm3btvQTr5PMpcmt8tbpE4sSy+Qiip8UR3ESyWViSiY
```

BAUlk1BwEJadEwmB68YRBCsPO1++/fP+2MP3EoFY/a7OV82qzGVio0Ad//BozFKoC8LRP69Ay
ORw7mPwsVH0GgGWX/x0ATJdm15ZuoR/z0tzBXbtZ6JovilcbC6uX34I9WVZZUUGZfuTlr1xG/
+/ZGTQ19+fTiiRfDTv10KfWNbACaqogESpajH60g89mJLmCl1bAr8ZAUf5Stl6nBrjVk8c0e7
lmBVzIUQYnfupHejfajjAamhyPfXgDG0CcWg24DWGyoGDp0ZD2KDVsXa1c6D097MpR3u4mS+8
uQWy0Bi68eD2kiN+9Gj0HTPsy0FPLhSpblrnGHrC6zUBny8C921koBTBW4a+bVblk3vEMNAFJ
iqGXgYUxAg+E2UXcz4PGTfhrjzeXQNiJSNvsPNhgTcwFr1VZDEDKcjOurgl4MO/OI1WrxmK6A
P+yK5wIaQ8fAg7EG9w3GBmKDNQAp+L0fFfsmdVSIFS/Ej4Y7H441N8EmY40InS02PwfvHczZd
XkWNwtQDJ3HEe48lo+fJyq7r9xJTas4MwXwf3pt2Y1D7bT27FMc57MqjeM2nX5PoUKnR3lQVc
H/lrh2Up2bLbz1Gn82aKwmHjwxX16iysG2c3UjCLCzW59L62lpMt2cfuooSnkwsnZ5ojCsVU
40mg04Unxkx0oRFnPRcw5CqV0RNeA5nNsdSZEpfJF6eVCIVHCL6SCL82CL/ZCbX9/AWSLiJoeb
FAeAAAAAE1FTkSuQmCC",

```
    "listContent": []
  }
  ,{
    "fieldName": "NETOS",
    "fieldValue": "",
    "base64": "",
    "listContent": [
      [
        {
          "fieldName": "CAMPOLL",
          "fieldValue": "LLVALOR1",
          "base64": "",
          "listContent": []
        }
      ],
      [
        {
          "fieldName": "CAMPOLL",
          "fieldValue": "LLVALOR2",
          "base64": "",
          "listContent": []
        }
      ]
    ]
  }
]
],[
{
  "fieldName": "CAMP01",
  "fieldValue": "3333",
  "base64": "",
  "listContent": []
}
,{
  "fieldName": "CATEGORIA",
  "fieldValue": "2",
```




Em caso de dúvidas,
entre em contato:

(11) 3474-1790
suporte@solucoes.wf