A3ES

Agência de Avaliação
e Acreditação
do Ensino Superior

SI A3ES – REST API

Guia de integração

Conteúdos |

PREFÁCIO **5**
Prefácio 1. | Âmbito e Propósito do Documento 5
Prefácio 2. | Abreviaturas e Acrónimos 5
TECNOLOGIA USADA **6**
Capítulo 1.1. | OpenAPI 6
Capítulo 1.2. | Swagger™ 6
Capítulo 1.3. | REST 6
Capítulo 1.4. | JSON 6
INTEGRAÇÃO **7**
Capítulo 2.1. | Fluxo 7
Tipos de operação: 7
Capítulo 2.2. | Pedido de dados de acesso 8
Capítulo 2.3. | Autorização 8
Capítulo 2.4. | Webservices 9
Capítulo 2.5. | Endpoints 12
Capítulo 2.5.1 | Serviço de obtenção de token 12
Capítulo 2.5.1 | Swagger UI 12
EXEMPLOS **13**
Capítulo 3.1. | Exemplo de pedido de dados de acesso 13
Capítulo 3.2. | Exemplo inserção de ficha de unidade curricular 14
Capítulo 3.3. | Exemplo inserção de ficha de docente 16

Identificação do Documento

Título: SI A3ES REST API – Guia de Integração

Língua do Documento: PT

Versão: 1.1

Palavras-Chave: SI A3ES, API, Guia de integração, REST

Data: 29/01/2024

Histórico de versões

Nº Versão	Data	Comentários
1.0	15/12/2023	Versão inicial;
1.1	29/01/2024	Uniformização das ligações dos *endpoints*; Correções;

****

NOTA: Todo o conteúdo inserido, neste manual, reproduz a versão do módulo no momento da sua publicação, e poderá sofrer eventuais alterações, por motivos de melhoria do produto ou por quaisquer outras razões que assim o justifiquem, sem aviso prévio ao cliente

Prefácio

Prefácio 1. | Âmbito e Propósito do Documento

No âmbito da integração com a SI A3ES REST API, pretende-se com este documento fornecer às instituições um guião sobre os passos necessários por forma a adicionar, alterar e remover fichas curriculares de docentes e fichas de unidades curriculares de forma automática nos relatórios de autoavaliação de ciclos de estudos em funcionamento e pedidos de acreditação prévia a novos ciclos de estudos.

A informação sobre os guiões e outras informações relevantes sobre os guiões a preencher pelas instituições estão disponíveis no sítio da A3ES na área Guiões e Procedimentos.

Prefácio 2. | Abreviaturas e Acrónimos

SI A3ES – Sistema Integrado Agência de Avaliação e Acreditação do Ensino Superior

REST - Representational State Transfer

IES – Instituição de Ensino Superior.

Tecnologia usada

Capítulo 1.1. | OpenAPI

A OpenAPI Specification (OAS) define uma descrição de interface padrão e independente de linguagem de programação para APIs HTTP, que permite que humanos e computadores descubram e compreendam os recursos de um serviço sem exigir acesso ao código-fonte, documentação adicional ou inspeção do tráfego de rede . Quando definido corretamente via OpenAPI, um consumidor pode entender e interagir com o serviço remoto com uma quantidade mínima de lógica de implementação. Semelhante ao que as descrições de interface fizeram para programação de nível inferior, a especificação OpenAPI remove suposições ao chamar um serviço.

Para mais informações, consultar o sítio em https://swagger.io/specification/ .

Capítulo 1.2. | Swagger™

Swagger™ é um projeto usado para descrever e documentar APIs RESTful. A especificação Swagger define um conjunto de arquivos necessários para descrever tal API. Esses arquivos podem ser usados pelo projeto Swagger-UI para exibir a API e Swagger-Codegen para gerar clientes em vários idiomas. Utilitários adicionais também podem tirar proveito dos arquivos resultantes, como ferramentas de teste.

O SI A3ES REST API usa a versão 2.0 do Swagger™. Para mais informação, consultar sítio em https://swagger.io/

Capítulo 1.3. | REST

REST, ou REpresentational State Transfer, é um estilo de arquitetura para fornecer padrões entre sistemas de computador na web, facilitando a comunicação entre os sistemas. Os sistemas compatíveis com REST, geralmente chamados de sistemas RESTful, são caracterizados por serem “stateless” e por separar as preocupações do cliente e do servidor. Abordaremos o que esses termos significam e por que são características benéficas para serviços na Web.

Para mis informação, consultar sítio em https://www.w3.org/2001/sw/wiki/REST.

Capítulo 1.4. | JSON

JavaScript Object Notation (JSON) é um formato padrão baseado em texto para representar dados estruturados com base na sintaxe de objeto JavaScript. É comumente usado para transmitir dados em aplicações web (por exemplo, enviar alguns dados do servidor para o cliente, para que possam ser exibidos em uma página web ou vice-versa). Para integração com a SI A3ES API recomendamos o uso de JSON. Embora sejam suportados XML e JSON na nossa implementação, no que respeita a integração com o Swagger. Para comunicações de autorização apenas é usado JSON.

Para mais informações, consultar sítio em https://www.json.org/.

Integração

Capítulo 2.1. | Fluxo

A integração com a API passa pelo passo de autorização em que é feita uma chamada para pedir o token e de seguida as chamadas às operações de GET, POST, PUT, DELETE à camada de webservices em REST. Os dados são transportados no formato JSON.

Tipos de operação:

- GET – Consular informação
- POST – Criar novos elementos
- PUT – Atualizar elementos Existentes
- DELETE – Apagar um determinar elemento

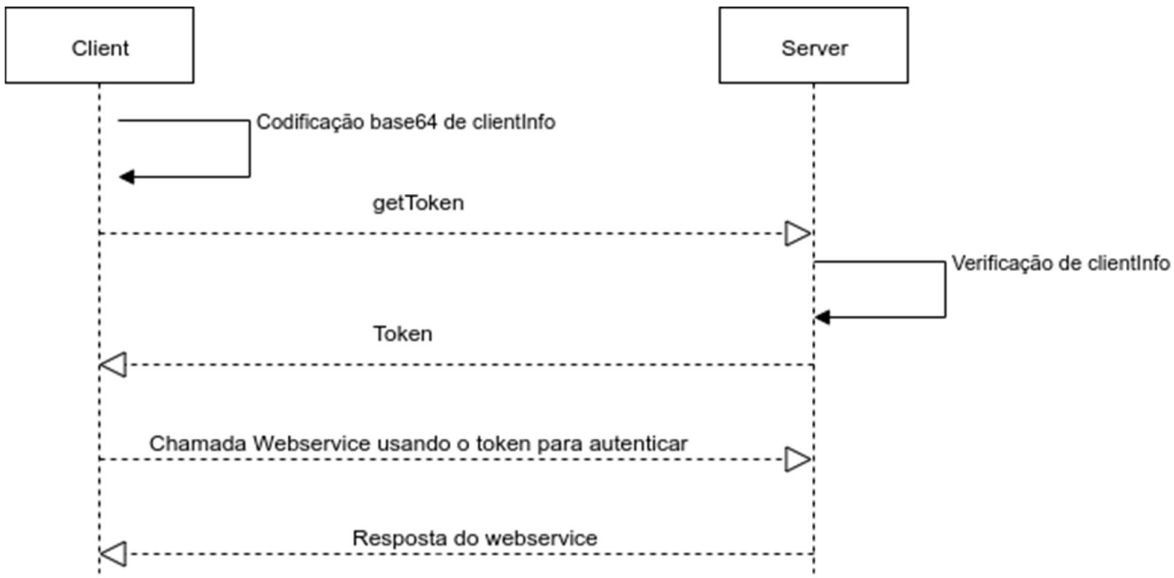

Imagem 1 – Fluxo de comunicação

Capítulo 2.2. | Pedido de dados de acesso

Para aceder à API dos webservices será necessário antes a obtenção dos dados de autenticação para acesso ao serviço para tal acontecer é necessário o envio de um email através da conta do responsável da instituição ou com o conhecimento do responsável da instituição para o email sia3es@a3es.pt com os seguintes dados:

- **Intituição/Unidade Orgânica:** Indicar o nome da IES ou da Unidade orgânica.
- **Perfil instituição:** indicar perfil da instituição, IES / UO. (O perfil de instituição IES poderá fazer consultar e alterar todos os processos da IES o perfil da instituição UO apenas poderá alterar os processos da unidade orgânica).
- **Email responsável pelo webservices:** indicar um email para o responsável pelo envio dos dados (será para este email que o sistema enviará os dados de acesso aos webservices).
- **IPs Produção:** lista de IPs válidos a submeterem os dados para a A3ES, em notação CIDR (separados por vírgulas caso exista mais do que uma gama de IPs),e.g., xxx.xxx.xxx.xxx/30, yyy.yyy.yyy.yyy/24.
- **IPs Testes:** lista de IPs válidos a submeterem os dados para a A3ES, em notação CIDR (separados por vírgulas caso exista mais do que uma gama de IPs),e.g., xxx.xxx.xxx.xxx/30, yyy.yyy.yyy.yyy/24.

Após o envio de email a A3ES irá inserir os dados no sistema de informação e deverá no email do responsável pelos webservices receber dois emails com os dados de acesso ao sistema de pré-produção e com os dados de acesso ao sistema de produção da A3ES.

Capítulo 2.3. | Autorização

Qualquer chamada feita a uma operação dos webservices requer a obtenção de um token com um pedido ao servidor. Para tal será necessário efetuar um "post" ao serviço "https://url/sia3es-rest-api/ajax/validationData/token", sendo que terá de ser enviado no header o atributo "clientInfo" que terá como conteúdo o json que contém o cliente, secret e o time em milissegundos. Essa informação terá de ser codificada em Base64.

Pode ser feita a validação se um token ainda é válido através da chamada ao serviço https://url/sia3es-rest-api/ajax/validationData/validateToken. No entanto não é garantido que no momento da chamada ao método da API, ainda esteja válido.

Exemplo do valor JSON antes de ser codificado em base64:

{"CLIENT":"clienttest","SECRET":"clientSecret","TIME":"1659692460593"}

Exemplo do valor a enviar após ser codificado em base64:

eyJUSU1FIjoiMTY1OTY5MjQ2MDU5MyIsIkNMSUVOVCI6ImNsaWVudHRlc3QiLCJTRUNSRVQiOiJjbGllbnRTZWNyZXQifQ==

Exemplo de código Java para esta operação:

```
url = new URL("https://url/sia3es-rest-api/ajax/validationData/token");
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("POST");
```

```
conn.setRequestProperty("accept", "application/json");
conn.setDoOutput(true);

HashMap<String, String> mapa = new HashMap<>();
mapa.put("CLIENT", client);
mapa.put("SECRET", secret);
mapa.put("TIME", Calendar.getInstance().getTimeInMillis() + "");
String json = JSONUtils.mapToJson(mapa);
String encoded = Base64.encodeBase64String(json.getBytes());

conn.setRequestProperty("clientInfo", encoded);

int responseCode = conn.getResponseCode();
```

O servidor irá devolver a resposta em **JSON** que terá a propriedade "**success**" com os valores "**true**" em caso de sucesso ou "false" em caso de falha. Em caso de sucesso, o token será devolvido na propriedade "**result**". Em caso de falha será devolvida a propriedade "**difexception**" que terá normalmente o valor "**unauthorized**".

Exemplo de uma resposta com sucesso:

{"result":"eyJFTlRJREFERVMiOiIxLDIiLCJSQURPTV9OVU1CRVJfMSI6IjU0MDI2NDg1Nzk3NDYwNjM3MjAiLCJUSU1FIjoiMTY1OTY5MzM0NDUwOSIsIlRZUEUiOiJUT0tFTiJ9","success":true}

Exemplo de uma resposta com falha:

{"difexception":"unauthorized","success":false}

É também importante referir que um token tem uma duração de alguns minutos, pelo que será recomendado efetuar o seu pedido a cada chamada aos webservices.

Capítulo 2.4. | Webservices

Como foi referido anteriormente foi usado o Swagger como interface para os webservices. Poderá ser consultada informação sobre cada operação acedendo ao Swagger UI da SIA3ES REST API, disponível em **https://url/sia3es-rest-api/**. No Swagger UI para além de estarem listados e documentados os métodos, também é possível consultar o modelo de dados.

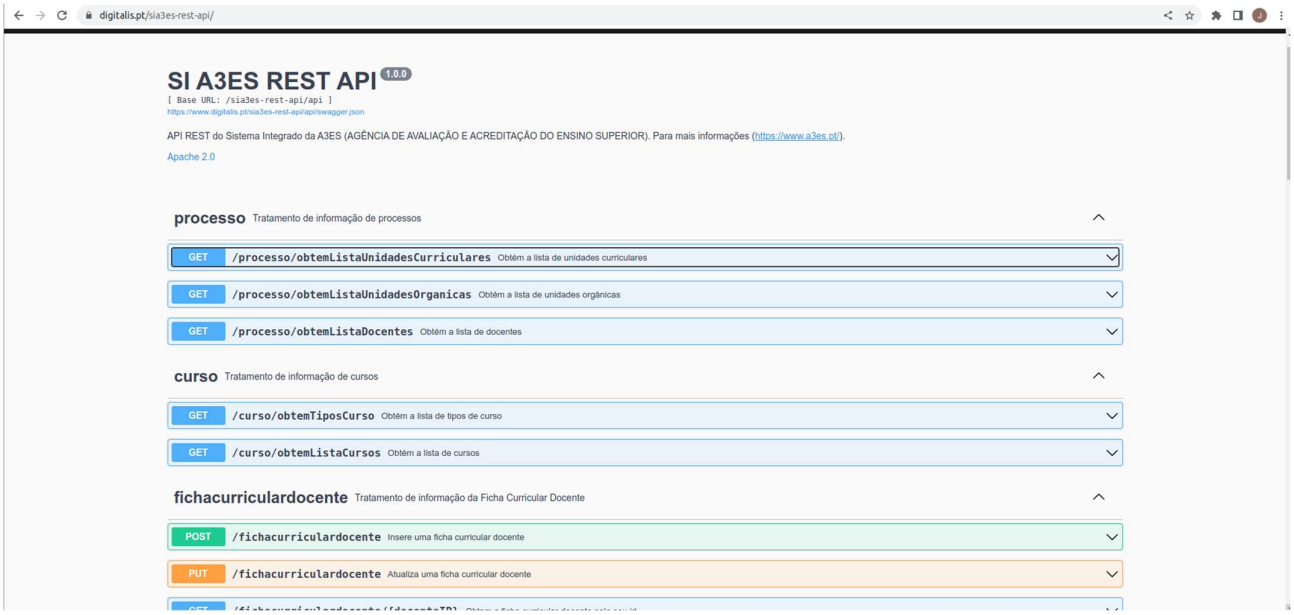

Imagem 2- SIA3ES REST API Swagger UI

Cada IES irá escolher a forma como quer implementar o cliente para acesso a esta camada de webservices. No entanto, como informação útil poderá experimentar gerar um cliente automaticamente para várias linguagens de programação no sítio https://editor.swagger.io/. Para tal, bastará fornecer a especificação do Swagger UI da A3ES REST API que poderá obter no endereço https://url/sia3es-rest-api/api/swagger.json. É da total responsabilidade da IES a escolha de como será implementado o código de acesso à API.

Na imagem abaixo vemos o Swagger Editor. No lado esquerdo deve ser feito "paste" da espeficação obtida em https://url/sia3es-rest-api/api/swagger.json. Carregar na opção "Generate Cliente" e escolher a linguagem de programação para a qual pretende gerar.

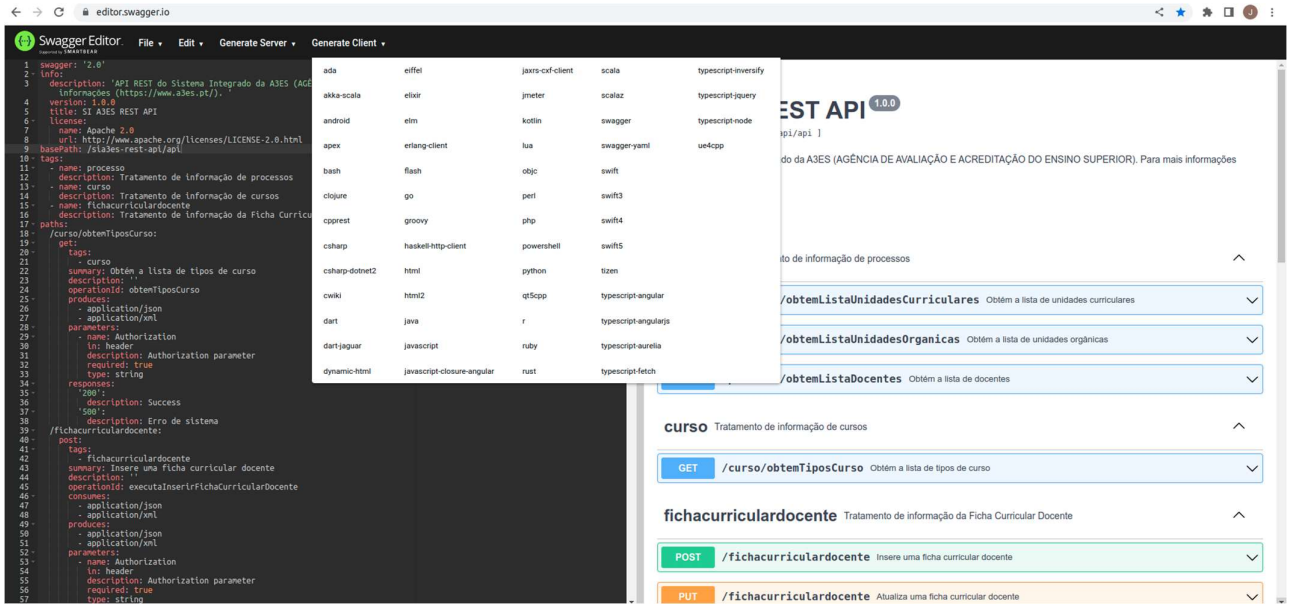

Imagem 3- Swagger editor

Cada chamada a operações da camada de webservice deverão ter o header “Authorization” que deverá ter como valor o **token** obtido no passo de autorização. Poderá testar estas chamadas no Swagger UI:

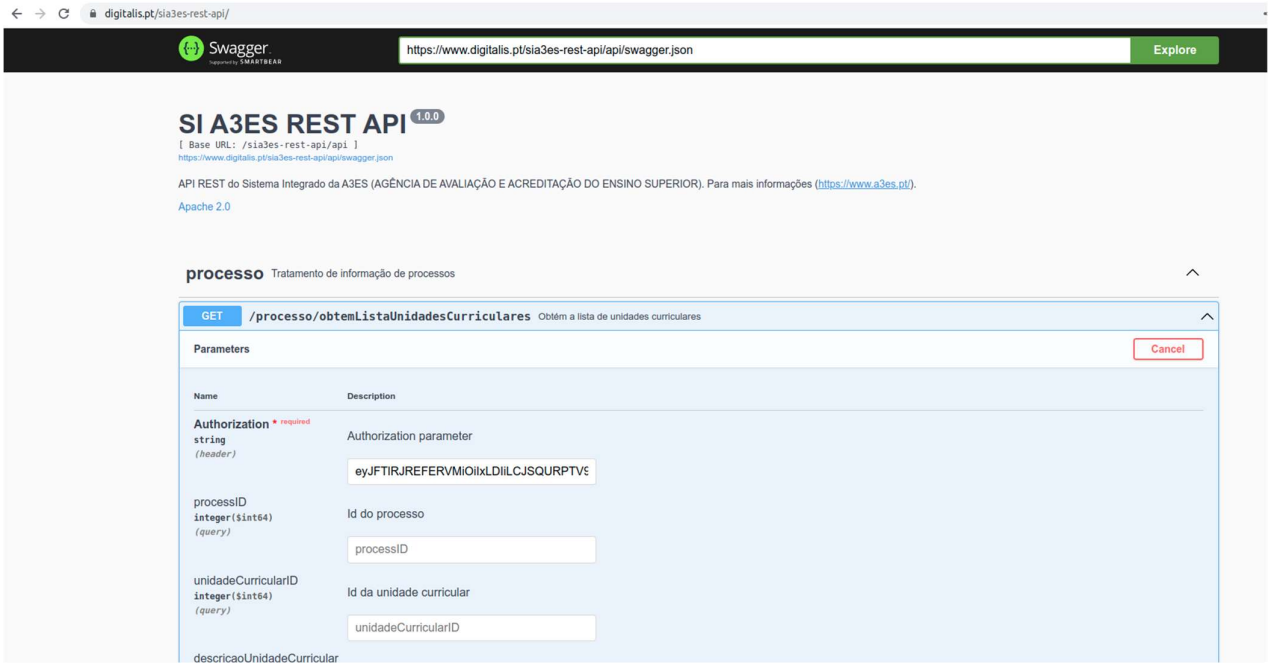

Imagem 4- Teste de uma chamada à operação /processos/obtemListaUnidadesCurriculares

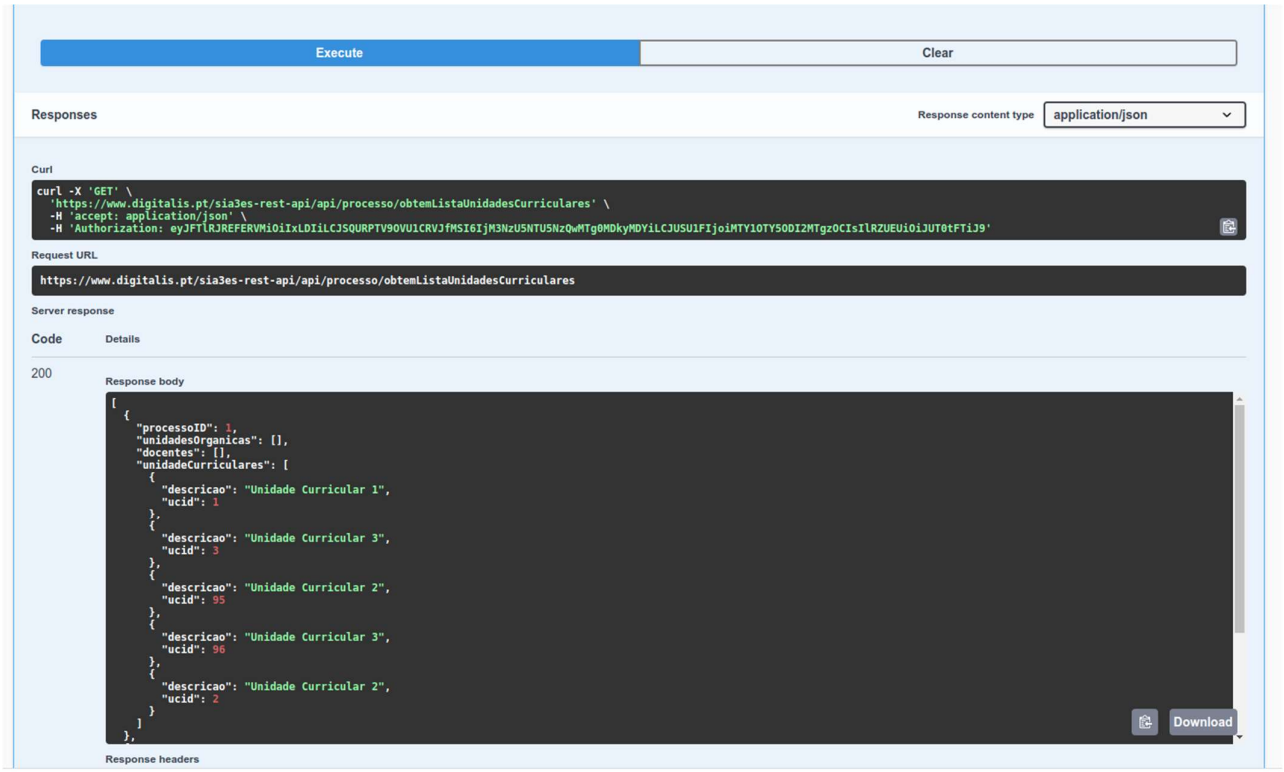

Imagem 5- Teste de uma chamada à operação /processos/obtemListaUnidadesCurriculares (continuação)

Capítulo 2.5. | Endpoints

Considerar que nos endpoints indicados os url-Préprodução e url-Produção serão enviados com o pedido inicial de acesso aos webservices.

Capítulo 2.5.1 | Serviço de obtenção de token

Pré-Produção	https://url-Prépodução/sia3es-rest-api/ajax/validationData/token
Produção	https://url-Produção/sia3es-rest-api/ajax/validationData/token

Capítulo 2.5.1 | Swagger UI

Pré-Produção	https://url-Prépodução/sia3es-rest-api/
Produção	https://url-Produção/sia3es-rest-api/

Exemplos

Nos exemplos seguintes vamos considerar que temos as seguintes instituições

- **Instituição A-** (ID 111)
  - Responsável da instituição: – ries@a1.exemplo.pt
  - Responsável pelo webservices: wsa1@a1.exemplo.pt

  - **UO1** (ID 1111)
    - Responsável pelo webservices: wsuo1@a1.exemplo.pt
  - **UO2** (ID 1112)
    - Responsável pelo webservices: wsuo2@a1.exemplo.pt

Considerar que
{{serverURL}} = https://url/sia3es-rest-api/

Capítulo 3.1. | Exemplo de pedido de dados de acesso

A partir da conta do RIES envia para sia3es@a3es.pt os seguintes dados

Intituição/Unidade Orgânica: Instituição A
Perfil instituição: IES
Email responsável pelo webservices: wsa1@a1.exemplo.pt
IPs Produção: xxx.xxx.xxx.xxx/30
IPs Testes: yyy.yyy.yyy.yyy/24.

Intituição**/Unidade Orgânica:** UO 1
Perfil instituição: UO
Email responsável pelo webservices: wsuo1@a1.exemplo.pt
IPs Produção: xxx.xxx.xxx.xxx/30
IPs Testes: yyy.yyy.yyy.yyy/24.

Os serviços da A3ES irão configurar a instituição para acederem aos dados e cada um dos responsáveis pelos webservices devem receber o seguinte email com os dados necessário para pedir o token de autorização:

Entidade: Instituição A
Client ID: 111
Secret: 4q+B^J=O
IP White list: xxx.xxx.xxx.xxx/30

Capítulo 3.2. | Exemplo inserção de ficha de unidade curricular

Fazer autenticação

GET {{serverURL}}/ajax/validationData/token

Fazer a autenticação com os dados enviados no exemplo anterior e seguindo os passos especificados no capítulo 2.3, o token devolvido terá de ser enviado no header de todas as próximas chamadas.

Obter lista de processos

GET {{serverURL}}/api/processo/obtemListaProcessos

Obter o campo processoID e cursoID do processo que se pretende alterar.

Inserir ficha

POST {{serverURL}}/api /fichaunidadecurricular

Submete a ficha de unidade curricular enviada exemplo enviado de seguida

Na página seguinte é mostrado um exemplo de uma ficha curricular para submissão.

```
{
	"areaCientificaUC": [
	],
	"avaliacao": "teste avaliacao",
	"avaliacaoEn": "test avaliacaoEn",
	"bibliografia": "teste bilbiografia",
	"bibliografiaEn": "test bilbiografiaen",
	"coerenciaConteudos": "teste coerenciaConteudos",
	"coerenciaConteudosEn": "test coerenciaConteudosEn",
	"coerenciaMetodologias": "teste coerenciaMetodologias",
	"coerenciaMetodologiasEn": "test coerenciaMetodologiasEn",
	"conteudos": "teste conteudos",
	"counteudosEn": "test conteudosEn",
	"processoID":3608,
	"cursoID": 3762,
	"descricao": "Teste UC",
	"descricaoEn": "Test UC",
	"docentesUC": [],
	"ects": 1.0,
	"horasTipoEnsino": [
		{
			"horasTipoAssincrono": null,
			"horasTipoPresencial": 33.3,
			"horasTipoSincrono": null,
			"tipoEnsino": {
				"id": 4
			}
		},
		{
			"horasTipoAssincrono": 10.3,
			"horasTipoPresencial": 20.3,
			"horasTipoSincrono": 30.0,
			"tipoEnsino": {
				"id": 1
			}
		}
	],
	"horasTrabalhoTotais": "1500",
	"metodologias": "teste metodologias",
	"metodologiasEn": "test metodologiasEn",
	"objetivos": "teste objetivos",
	"objetivosEn": "teste objetivosEn",
	"observacoes": null,
	"observacoesEn": null,
	"tipo": "N",
	"tipoDuracao": {
		"id": 12
	},
	"unidadesCurricularesOpcao": []
}
```

Capítulo 3.3. | Exemplo inserção de ficha de docente

Fazer autenticação

GET {{serverURL}}/ajax/validationData/token

Fazer a autenticação com os dados enviados no exemplo anterior e seguindo os passos especificados no capítulo 2.3, o token devolvido terá de ser enviado no header de todas as próximas chamadas.

Obter lista de processos

GET {{serverURL}}/api/processo/obtemListaProcessos

Obter o campo processoID e cursoID do processo que se pretende alterar.

Inserir ficha

POST {{serverURL}}/api /fichacurriculardocente

Submete a ficha de unidade curricular enviada exemplo enviado de seguida

De seguinda é mostrado um exemplo JSON de uma ficha curricular de docente para submissão.


```
{
  "anoEspecialista":null,
  "anoGrau":2000,
  "areaCientificaEspecialista":null,
  "areaCientificaEspecialistaEn":null,
  "areaCientificaGrau":"Teste areaCientificaGrau",
  "areaCientificaGrauEn":"Teste areaCientificaGrauEN",
  "categoriaDocente":{"id":4},
  "cienciavitae":"1111-1111-1111",
  "distribuicaoServicoDocente":[
  ],
  "docenteResponsavel":true,
  "email":"docenteteste@a1.exemplo.com",
  "especialista":0,
  "estadoFichaDocente":"PN",
  "instituicaoGrau":"Instituicao A",
  "instituicaoGrauEn":"Instituicao A",
  "nome":"Docente teste",
  "orcid":null,
  "outrosGrausDocente":[
  ],
  "processID":1111,
  "cursoID": 1111,
  "regimeTempo":50,
  "tipoCurso":{
     "id":2
  },
  "uinvestigacaoDocente":[
  ],
  "vinculoDocente":{
      "id":1
  }
}
```