

Doc API REST FUTAPP

Usuário Service	1
Login de usuário	1
Renovar autenticação	2
Organizador Service	3
List	3
Post	4
Get	5
Update	6
Delete	7
Campeonato Service	8
List	8
Post	9
Get	10
Update	11
Delete	12
Clube Service	13
List	13
Post	14
Get	15
Update	16
Delete	17
Jogador Service	18
List	18
Post	19
Get	20
Update	21
Delete	22
Jogo Service	23
List	23
Post	24
Get	25
Update	26
Delete	27

1. Usuário Service

WebService dedicado a ações relacionadas a operações comuns a todos os tipos de usuários.

Path: `/api-token-auth/`

1. Login de usuário

Descrição	Método para realizar o login e obter o token. O token gerado será retornado via Header da requisição HTTP.
Path	<code>/api-token-auth/</code>
Method	POST
URL Parameters	
Data	<pre>{ "username": "", "password": "" }</pre>
Return type	Void
Return data	<code>header.payload.signature</code>
Authority	No restrictions

2. Renovar autenticação

Descrição	Método para realizar a renovação do token. O token gerado será retornado via Header da requisição HTTP.
Path	/api-token-refresh/
Method	POST
URL Parameters	
Data	<code>{"token": "<EXISTING_TOKEN>"}</code>
Return type	Void
Return data	<code>header.payload.signature</code>
Authority	No restrictions

2. Organizador Service

WebService dedicado a ações relacionadas a operações de CRUD aos dados do sistema referente a Organizador.

Path: **/organizador/**

1. List

Descrição	Método para pegar a lista de dados. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	/organizador
Method	GET
URL Parameters	
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx", "campeonatos": [{ "id": x, "nome": "xxx", "numero_times": x, "imagem": "xxx", "regulamento": "xxx" }] }]</pre>
Authority	Usuário

2. Post

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/organizador
Method	POST
URL Parameters	
Data	<pre>{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx" }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx", "campeonatos": [] }]</pre>
Authority	Usuário

3. Get

Descrição	Método para pegar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/organizador</code>
Method	GET
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja pesquisar
Data	
Return type	Void
Return data	<pre>{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx", "campeonatos": [{ "id": x, "nome": "xxx", "numero_times": x, "imagem": "xxx", "regulamento": "xxx" }] }</pre>
Authority	Usuário

4. Update

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/organizador
Method	PUT
URL Parameters	<ul style="list-style-type: none">• /'{id}' = ID do registro que se deseja alterar
Data	<pre>{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx", }</pre>
Return type	Void
Return data	<pre>{ "id": xxx, "nome": "xxx", "sede": "xxx", "presidente": "xxx", "fundacao": "xxx-xx-xx", "imagem": "xxx", "campeonatos": [] }</pre>
Authority	Usuário

5. Delete

Descrição	Método para deletar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/organizador</code>
Method	DELETE
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja deletar
Data	
Return type	Void
Return data	
Authority	Usuário

3. Campeonato Service

WebService dedicado a ações relacionadas a operações de CRUD aos dados do sistema referente a Campeonato.

Path: **/campeonato/**

1. List

Descrição	Método para pegar a lista de dados. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	/campeonato
Method	GET
URL Parameters	
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx, "clubes": [{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "escudo": "xxx", "hino": "xxx" }] }]</pre>
Authority	Usuário

2. Post

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/campeonato
Method	POST
URL Parameters	
Data	<pre>{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx, "clubes": [] }]</pre>
Authority	Usuário

3. Get

Descrição	Método para pegar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/campeonato</code>
Method	GET
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja pesquisar
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx, "clubes": [{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "escudo": "xxx", "hino": "xxx" }] }]</pre>
Authority	Usuário

4. Update

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/campeonato
Method	PUT
URL Parameters	<ul style="list-style-type: none">• /'{id}' = ID do registro que se deseja alterar
Data	<pre>{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "numero_times": xxx, "imagem": "xxx", "regulamento": "xxx", "organizador": xxx, "clubes": [] }]</pre>
Authority	Usuário

5. Delete

Descrição	Método para deletar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/campeonato</code>
Method	DELETE
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja deletar
Data	
Return type	Void
Return data	
Authority	Usuário

4. Clube Service

WebService dedicado a ações relacionadas a operações de CRUD aos dados do sistema referente a Clube.

Path: **/clube/**

1. List

Descrição	Método para pegar a lista de dados. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token>
Path	/clube
Method	GET
URL Parameters	
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", "jogadores": [{ "id": xxx, "nome": "xxx", "idade": xxx, "time": xxx, "peso": xxx, "foto": "xxx" }] }]</pre>
Authority	Usuário

2. Post

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/clube
Method	POST
URL Parameters	
Data	<pre>{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", "jogadores": [] }]</pre>
Authority	Usuário

3. Get

Descrição	Método para pegar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/clube</code>
Method	GET
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja pesquisar
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", "jogadores": [{ "id": xxx, "nome": "xxx", "idade": xxx, "time": xxx, "peso": xxx, "foto": "xxx" }] }]</pre>
Authority	Usuário

4. Update

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/clube
Method	PUT
URL Parameters	<ul style="list-style-type: none">• /'{id}' = ID do registro que se deseja alterar
Data	<pre>{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "criacao": "xxx", "pontuacao": xxx, "campeonato": xxx, "escudo": "xxx", "hino": "xxx", "jogadores": [] }]</pre>
Authority	Usuário

5. Delete

Descrição	Método para deletar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/clube</code>
Method	DELETE
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja deletar
Data	
Return type	Void
Return data	
Authority	Usuário

5. Jogador Service

WebService dedicado a ações relacionadas a operações de CRUD aos dados do sistema referente a Jogador.

Path: **/jogador/**

1. List

Descrição	Método para pegar a lista de dados. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	/jogador
Method	GET
URL Parameters	
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "creat_at": "xxx", "update_at": "xxx", "time": xxx }]</pre>
Authority	Usuário

2. Post

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/jogador
Method	POST
URL Parameters	
Data	<pre>{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "time": xxx }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "creat_at": "xxx", "update_at": "xxx", "time": xxx }]</pre>
Authority	Usuário

3. Get

Descrição	Método para pegar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/jogador</code>
Method	GET
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja pesquisar
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "creat_at": "xxx", "update_at": "xxx", "time": xxx }]</pre>
Authority	Usuário

4. Update

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/jogador
Method	PUT
URL Parameters	<ul style="list-style-type: none">• /'{id}' = ID do registro que se deseja alterar
Data	<pre>{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "time": xxx }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "nome": "xxx", "idade": xxx, "peso": xxx, "foto": "xxx", "creat_at": "xxx", "update_at": "xxx", "time": xxx }]</pre>
Authority	Usuário

5. Delete

Descrição	Método para deletar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/jogador</code>
Method	DELETE
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja deletar
Data	
Return type	Void
Return data	
Authority	Usuário

5. Jogo Service

WebService dedicado a ações relacionadas a operações de CRUD aos dados do sistema referente a Jogo.

Path: **/jogo/**

1. List

Descrição	Método para pegar a lista de dados. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	/jogo
Method	GET
URL Parameters	
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", "creat_at": "xxx", "update_at": "xxx" }]</pre>
Authority	Usuário

2. Post

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/jogo
Method	POST
URL Parameters	
Data	<pre>{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", "creat_at": "xxx", "update_at": "xxx" }]</pre>
Authority	Usuário

3. Get

Descrição	Método para pegar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/jogo</code>
Method	GET
URL Parameters	<ul style="list-style-type: none"><code>/{id}'</code> = ID do registro que se deseja pesquisar
Data	
Return type	Void
Return data	<pre>[{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", "creat_at": "xxx", "update_at": "xxx" }]</pre>
Authority	Usuário

4. Update

Descrição	Método para salvar um registro. O token do usuário logado deve ser enviado como HEADER, com o tipo Authorization: JWT <your_token> , e junto no HEADER o Content-Type: multipart/form-data para envio de Arquivos
Path	/jogo
Method	PUT
URL Parameters	<ul style="list-style-type: none">• /'{id}' = ID do registro que se deseja alterar
Data	<pre>{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", }</pre>
Return type	Void
Return data	<pre>[{ "id": xxx, "time_casa": xxx, "time_fora": xxx, "gol_tm_casa": xxx, "gol_tm_fora": xxx, "data": "xxx", "creat_at": "xxx", "update_at": "xxx" }]</pre>
Authority	Usuário

5. Delete

Descrição	Método para deletar um objeto específico. O token do usuário logado deve ser enviado como HEADER, com o tipo <code>Authorization: JWT <your_token></code>
Path	<code>/jogo</code>
Method	DELETE
URL Parameters	<ul style="list-style-type: none"><code>/{id}</code> = ID do registro que se deseja deletar
Data	
Return type	Void
Return data	
Authority	Usuário