

Benutzeranleitung der OpenAPE Javascript Library

Inhalt

Voraussetzung	2
Funktionen	3
1. User	3
1.1. createUser	3
1.2. getUser	4
2. InitializeLibrary	5
3. UserContext	6
3.1. createUserContext	6
3.2. getUserContext	7
3.3. updateUserContext	7
3.4. deleteUserContext	8
4. TaskContext	9
4.1. createTaskContext	9
4.2. getTaskContext	9
4.3. updateTaskContext	10
4.4. deleteTaskContext	10
5. EquipmentContext	11
5.1. createEquipmentContext	11
5.2. getEquipmentContext	11
5.3. updateEquipmentContext	12
5.4. deleteEquipmentContext	12
6. EnvironmentContext	14
6.1. createEnvironmentContext	14
6.2. getEnvironmentContext	14
6.3. updateEnvironmentContext	15
6.4. deleteEnvironmentContext	15

Voraussetzung

Um die Bibliothek nutzen zu können, ist es erforderlich, dass zusätzlich jQuery eingebunden wird. Es ist zu empfehlen jQuery vor der Bibliothek einzubinden.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
<script src="https://code.jquery.com/jquery-3.2.1.js"></script>
<script type="text/javascript" src="openape.js"></script>
</head>
<body>

</body>
</html>
```

Nachdem jQuery und die Bibliothek eingebunden wurde, können weitere Javascript-Dateien eingebunden werden, die auf die Funktionen der OpenAPE-Bibliothek zugreifen. Eine genaue Beschreibung der Funktionen kann aus dem Kapitel „Funktionen“ entnommen werden.

Funktionen

1. User

1.1. createUser

Beschreibung

Die Verwendung der *createUser()*-Methode ermöglicht es den Benutzern der Javascript-Bibliothek das Anlegen von neuen Benutzern, die erforderlich sind um mit Kontexten zu interagieren.

Aufruf

```
openape.createUser(username, email, password, serverurl);
```

Die Übergabe der Serverurl kann über drei verschiedene Arten erfolgen:

- Es wird keine Serverurl übergeben, dann wird die `http://openape.gpii.eu` Adresse verwendet, auf die die Bibliothek zugreifen wird.

Beispiel

```
openape.createUser("admin", "admin@gmail.com", "12345");
```

- Wenn ein `/` übergeben wird, dann wird automatisch die lokale Adresse verwendet, unter der die Bibliothek läuft. Diese Methode ist zu empfehlen, wenn die Bibliothek in einer lokalen OpenAPE-Anwendung integriert wurde.

Beispiel

```
openape.createUser("admin", "admin@gmail.com", "12345", "/" );
```

- Weiterhin ist es möglich eine zusätzliche Serverurl zu übergeben.

Beispiel

```
openape.createUser("admin", "admin@gmail.com", "12345", "http://localhost:8080");
```

Rückgabe

```
responseText: "Done! Your ID is 59563eddf4d797239429ac74"
status: 200
```

1.2. getUser

Beschreibung

Durch die Verwendung der *getUser()*-Methode können alle Informationen eines Benutzers abgefragt werden. Diesbezüglich ist es erforderlich, dass Die Bibliothek vorher initialisiert wurde.

Aufruf

```
openape.getUser();
```

Rückgabe

Bei einem erfolgreichen Aufruf der *getUser()*-Methode werden im *responseText* die zugehörigen Benutzerinformationen zurückgeliefert sowie 200 als Status.

```
responseText: "{"id":"59340c3d04bac129c09deed6","username":"admin",
               "email":"admin@gmx.de","roles":["admin"]}"
status: 200
```

2. InitializeLibrary

Beschreibung

Damit Kontexte abgefragt, angelegt, bearbeitet oder gelöscht werden können, ist es erforderlich, dass die OpenAPE Javascript-Bibliothek einmalig initialisiert wird. Zur Initialisierung der Bibliothek ist ein bereits angelegter Benutzer erforderlich und bei Bedarf eine Serverurl, auf die die Bibliothek zugreifen soll. Die Übergabe der Serverurl kann über drei verschiedene Arten erfolgen:

- Es wird keine Serverurl übergeben, dann wird die `http://openape.gpii.eu` Adresse verwendet, auf die die Bibliothek zugreifen wird.

Beispiel

```
openape.initializeLibrary("admin", "12345");
```

- Wenn ein `/` übergeben wird, dann wird automatisch die lokale Adresse verwendet, unter der die Bibliothek läuft. Diese Methode ist zu empfehlen, wenn die Bibliothek in einer lokalen OpenAPE-Anwendung integriert wurde.

Beispiel

```
openape.initializeLibrary("admin", "12345", "/");
```

- Weiterhin ist es möglich eine zusätzliche Serverurl zu übergeben.

Beispiel

```
openape.initializeLibrary("admin", "12345", "http://localhost:8080");
```

Rückgabe

Wenn die Bibliothek erfolgreich initialisiert werden konnte, dann wird im `responseText` der Access Token mit der Ablaufzeit zurückgegeben und einem 200 Status

`responseText:` `{"access_token":"<token>","expires_in":"1440"}`

`status:` `200`

3. UserContext

3.1. createUserContext

Beschreibung

Das Anlegen von einem *user-context* erfolgt durch die Verwendung der *createUserContext()*-Methode.

Aufruf

```
openape.createUserContext(JSONPayload);
```

Beispiels JSON-Payload

```
{"contexts":{"default":{"name":"Default preferences","preferences":{"http://registry.gpii.net/common/magnifierEnabled":"false","http://registry.gpii.net/applications/org.chrome.cloud4chrome/invertColours":"false"},"dark":{"conditions":{"type":"and","operands":[{"type":"ge","operands":["http://registry.gpii.net/common/env/visual.luminance","0"]},{"type":"le","operands":["http://registry.gpii.net/common/env/visual.luminance","200"]}}]},"name":"little environmental light","preferences":{"http://registry.gpii.net/common/magnifierEnabled":"true","http://registry.gpii.net/common/magnification":"2"}}}}
```

Rückgabe

Wenn ein *user-context* erfolgreich angelegt werden konnte, dann wird als *responseText* die ID des angelegten Kontextes angezeigt sowie der Status 201.

```
responseText: "59564488f4d797239429ac75"
```

```
status: 201
```

3.2. getUserContext

Beschreibung

Durch die Verwendung der *getUserContext()*-Methode können Informationen zu einem bestimmten *user-context* abgefragt werden.

Aufruf

```
openape.getUserContext(userContextId);
```

Beispiel

```
openape.getUserContext("594d415d04bac1218898124d");
```

Rückgabe

Wenn die Abfrage erfolgreich durchgeführt werden konnte, dann wird im *responseText* der jeweilige *user-context* mit der ID des Benutzers sowie dem Status 200 angezeigt.

```
responseText:  {"owner":"59340c3d04bac129c09deed6","contexts":{"default":{"name":"Default preferences","preferences":{"http://registry.gpii.net/common/magnifierEnabled":"false","http://registry.gpii.net/applications/org.chrome.cloud4chrome/invertColours":"false"},"dark":{"conditions":{"type":"and","operands":[{"type":"ge","operands":["http://registry.gpii.net/common/env/visual.luminance","0"]},{"type":"le","operands":["http://registry.gpii.net/common/env/visual.luminance","200"]}]},"name":"little environmental light","preferences":{"http://registry.gpii.net/common/magnifierEnabled":"true","http://registry.gpii.net/common/magnification":"2"}}},"public":false}"}
status:        200
```

3.3. updateUserContext

Beschreibung

Die *updateUserContext()*-Methode ist dafür zuständig bereits angelegte Benutzerkontexte nachträglich bearbeiten zu können.

Aufruf

```
openape.updateUserContext(userContextId, JSONPayload);
```

3.4. deleteUserContext

Beschreibung

Ein *user-context* kann durch die Verwendung der *deleteUserContext()*-Methode gelöscht werden. Diesbezüglich ist es erforderlich, dass eine *user context Id* als Parameter übergeben wird.

Aufruf

```
openape.deleteUserContext(userContextId);
```

Beispiel

```
openape.deleteUserContext("594d415d04bac1218898124d");
```

Rückgabe

```
responseText:    ""
status:          204
```


4. TaskContext

4.1. createTaskContext

Beschreibung

Das Anlegen von einem *task-context* erfolgt durch die Verwendung der *createTaskContext()*-Methode.

Aufruf

```
openape.createTaskContext(JSONPayload);
```

Beispiels JSON-Payload

```
{"property": [{"name": "http://openurc.org/ns/res#generalTask", "value": "spreadsheet"}, {"name": "http://openurc.org/ns/res#fileName", "value": "Budget2016.xlsx"}, {"name": "http://openurc.org/ns/res#application", "value": "MS Excel"}]}
```

Rückgabe

Wenn ein *task-context* erfolgreich angelegt werden konnte, dann wird als *responseText* die ID des angelegten Kontextes angezeigt sowie der Status 201.

```
responseText: "595651f4f4d797239429ac76"
status: 201
```

4.2. getTaskContext

Beschreibung

Durch die Verwendung der *getTaskContext()*-Methode können Informationen zu einem bestimmten *task-context* abgefragt werden.

Aufruf

```
openape.getTaskContext(taskContextId);
```

Beispiel

```
openape.getTaskContext("595651f4f4d797239429ac76");
```

Rückgabe

Wenn die Abfrage erfolgreich durchgeführt werden konnte, dann wird im *responseText* der jeweilige *task-context* mit der ID des Benutzers sowie dem Status 200 angezeigt.

```
responseText:      "{"owner":"59340c3d04bac129c09deed6","property":  
[{"name":"http://openurc.org/ns/res#generalTask","value":"spreadsheeti","descriptors":{}},{  
"name":  
"http://openurc.org/ns/res#fileName","value":"Budget2016.xlsx","descriptors":{}},{  
"name":"http://o  
penurc.org/ns/res#application","value":"MS Excel","descriptors":{}}],"public":false}"  
status:           200
```

4.3. updateTaskContext

Beschreibung

Die *updateTaskContext()*-Methode ist dafür zuständig einen bereits angelegten *task-context* nachträglich bearbeiten zu können.

Aufruf

```
openape.updateTaskContext(taskContextId, JSONPayload);
```

Rückgabe

```
responseText:      ""  
status:           200
```

4.4. deleteTaskContext

Beschreibung

Ein *task-context* kann durch die Verwendung der *deleteTaskContext()*-Methode gelöscht werden. Diesbezüglich ist es erforderlich, dass eine *task context id* als Parameter übergeben wird.

Aufruf

```
openape.deleteTaskContext(taskContextId);
```

Rückgabe

```
responseText:      ""  
status:           204
```

5. EquipmentContext

5.1. createEquipmentContext

Beschreibung

Das Anlegen von einem *equipment-context* erfolgt durch die Verwendung der *createEquipmentContext()*-Methode.

Aufruf

```
openape.createEquipmentContext(JSONPayload);
```

Beispiels JSON-Payload

```
{ "property": [{"name": "http://openurc.org/ns/res#friendlyName", "value": "My iPad", "descriptors": [{"name": "http://www.w3.org/XML/1998/namespace/lang", "value": "en"}]}, {"name": "http://openurc.org/ns/res#friendlyName", "value": "Mein iPad", "descriptors": [{"name": "http://www.w3.org/XML/1998/namespace/lang", "value": "de"}]}, {"name": "http://openurc.org/ns/res#devicePlatform", "value": "iOS"}, {"name": "http://openurc.org/ns/res#deviceType", "value": "iPad-3gen"}, {"name": "http://openurc.org/ns/res#resolution", "value": "1536x2048"}]}
```

Rückgabe

Wenn ein *equipment-context* erfolgreich angelegt werden konnte, dann wird als *responseText* die ID des angelegten Kontexts angezeigt sowie der Status 201.

```
responseText: "59565b32f4d797239429ac77"
```

```
status: 201
```

5.2. getEquipmentContext

Beschreibung

Durch die Verwendung der *getEquipmentContext()*-Methode können Informationen zu einem bestimmten *equipment-context* abgefragt werden.

Aufruf

```
openape.getEquipmentContext(equipmentContextId);
```

Beispiel

```
openape.getEquipmentContext("594937a4196ebd1fd4a4fd87");
```

Rückgabe

Wenn die Abfrage erfolgreich durchgeführt werden konnte, dann wird im *responseText* der jeweilige *equipment-context* mit der ID des Benutzers sowie dem Status 200 angezeigt.

```
responseText:      "{"owner":"59340c3d04bac129c09deed6","propertys":  
[{"name":"http://openurc.org/ns/res#friendlyName","value":"My  
iPad","descriptors":{"name":"http://www.w3.org/XML/1998/namespace/lang","value":"en"}}, {"na  
me":"http://openurc.org/ns/res#friendlyName","value":"Mein  
iPad","descriptors":{"name":"http://www.w3.org/XML/1998/namespace/lang","value":"de"}}, {"na  
me":"http://openurc.org/ns/res#devicePlatform","value":"iOS","descriptors":{}}, {"name":"htt  
p://openurc.org/ns/res#deviceType","value":"iPad-  
3gen","descriptors":{}}, {"name":"http://openurc.org/ns/res#resolution","value":"1536x2048",  
"descriptors":{}}], "public":false}"  
status:           200
```

5.3. updateEquipmentContext

Beschreibung

Die *updateEquipmentContext()*-Methode ist dafür zuständig einen bereits angelegten *equipment-context* nachträglich bearbeiten zu können.

Aufruf

```
openape.updateEquipmentContext(equipmentContextId, JSONPayload);
```

Rückgabe

```
responseText:      ""  
status:           200
```

5.4. deleteEquipmentContext

Beschreibung

Durch die Verwendung der *deleteEquipmentContext()*-Methode kann ein bestimmter *equipment-context* gelöscht werden. Diesbezüglich ist es erforderlich, dass eine *equipment context id* als Parameter übergeben wird.

Aufruf

```
openape.deleteEquipmentContext(equipmentContextId);
```

Beispiel

```
openape.deleteEquipmentContext("594937a4196ebd1fd4a4fd87");
```

Rückgabe

```
responseText: ""
```

```
status: 204
```

6. EnvironmentContext

6.1. createEnvironmentContext

Beschreibung

Das Anlegen von einem *environment-context* erfolgt durch die Verwendung der *createEnvironmentContext()*-Methode.

Aufruf

```
openape.createEnvironmentContext(JSONPayload);
```

Beispiels JSON-Payload

```
{"property": [{"name": "http://registry.gpii.net/common/environment/visual.luminance", "value": "4"}, {"name": "http://terms.gpii.net/time-of-day", "value": "1905"}]}
```

Rückgabe

Wenn ein *environment-context* erfolgreich angelegt werden konnte, dann wird als *responseText* die ID des angelegten Kontextes angezeigt sowie der Status 201.

```
responseText: "595778f6196ebd25c087963c"
status: 201
```

6.2. getEnvironmentContext

Beschreibung

Durch die Verwendung der *getEnvironmentContext()*-Methode können Informationen zu einem *environment-context* abgefragt werden.

Aufruf

```
openape.getEnvironmentContext(environmentContextId);
```

Beispiel

```
openape.getEnvironmentContext("595778f6196ebd25c087963c");
```

Rückgabe

```
responseText:      "{\"owner\":\"59340c3d04bac129c09deed6\",\"property\":  
[{"name":"http://registry.gpii.net/common/environment/visual.luminance","value":"4","descriptors":{}},{  
{"name":"http://registry.gpii.net/time-of-day","value":"1905","descriptors":{}},{  
{"name":"http://registry.gpii.net/public","value":"false"}]}status:           200
```

6.3. updateEnvironmentContext

Beschreibung

Die `updateEnvironmentContext()`-Methode ist dafür zuständig einen bereits angelegten *environment-context* nachträglich bearbeiten zu können.

Aufruf

```
openape.updateEnvironmentContext(environmentContextId, JSONPayload);
```

Rückgabe

```
responseText:      ""  
status:           200
```

6.4. deleteEnvironmentContext

Beschreibung

Ein *environment-context* kann durch die Verwendung der `deleteEnvironmentContext()`-Methode gelöscht werden. Diesbezüglich ist es erforderlich, dass eine *environment context id* als Parameter übergeben wird.

Aufruf

```
openape.deleteEnvironmentContext(environmentContextId);
```

Beispiel

```
openape.deleteEnvironmentContext("595778f6196ebd25c087963c");
```

Rückgabe

```
responseText: ""  
status: 204
```