

Gamification

An API to implement gamification features in the swagger-2.0 specification.

Version 1.0.0

Contact information

Guillaume Serneels

guillaume.serneels@heig-vd.ch (mailto:guillaume.serneels@heig-vd.ch)

www.heig-vd.ch (http://www.heig-vd.ch)

License

MIT (http://opensource.org/licenses/MIT)

Paths

/badges

GET /badges

Summary

Returns every badges


Description

Returns every badges

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	\Leftrightarrow string

Responses

Code	Description	Schema
200	Badges returned	\Leftrightarrow  [badgeWithLocation { }]
403	Invalid token	

Try this operation

POST /badges

Summary

Creates a new badge

Description

Creates a new badge

Parameters

Name	Located in	Description	Required	Schema
badge	body	Badge object to add to the store	Yes	<div>↔<div>▼ badge { name: string * }</div></div>
token	header	token to be passed as a header	Yes	<div>↔string</div>

Responses

Code	Description	Headers	Type	Details
201	Badge created	<div><div>Name</div><div>location</div></div> <div><div>Description</div><div>location of the newly created badge,</div></div>	"string"	► Object
403	Invalid token			
422	Invalid payload			

Try this operation

/badges/{badgeId}

GET /badges/{badgeId}

Summary

Returns a single badge

Description

Returns a single badge

Parameters

Name	Located in	Description	Required	Schema
badgeId	path	ID of badge to fetch	Yes	⇔ integer (int64)
token	header	token to be passed as a header	Yes	⇔ string

Responses

Code	Description	Schema
200	Badge found	⇔ <div>▼ badge { name: string *</div>
403	Invalid token	
404	Badge not found	
422	Invalid payload	

Try this operation

PUT /badges/{badgeId}

Summary

Update an existing badge

Description

Update an existing badge

Parameters

Name	Located in	Description	Required	Schema
badge	body	Badge object to add to the store	Yes	\Rightarrow ▼ badge { name: string * }
badgeId	path	Id of the badge that needs to be updated	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
200	Badge updated
403	Invalid token
404	Badge not found
422	Invalid payload

[Try this operation](#)

DELETE /badges/{badgeId}

Summary

Delete an existing badge

Description

Delete an existing badge

Parameters

Name	Located in	Description	Required	Schema
badgeId	path	Id of the badge that needs to be deleted	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
200	Badge deleted successfully
403	Invalid token
404	Badge not found
422	Invalid payload

Try this operation

/pointScales

GET /pointScales

Summary

Returns every pointScales

Description

Returns every pointScales

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	⇔ string

Responses

Code	Description	Schema
200	pointScales list	⇔ <div>▼ [▶ pointScaleWithLocation { }]</div>
403	Invalid token	

Try this operation

POST /pointScales

Summary

Creates a new pointScale

Description

Creates a new pointScale

Parameters

Name	Located in	Description	Required	Schema
pointScale	body	pointScale object to add to the store	Yes	<div>↔</div> <div>▼pointScale { name: string * }</div>
token	header	token to be passed as a header	Yes	<div>↔</div> string

Responses

Code	Description	Headers			
		Name	Description	Type	Details
201	pointScale created	location	Location of the pointScale	"string"	► Object
403	Invalid token				
422	Invalid payload				

Try this operation

/pointScales/{pointScaleId}

GET /pointScales/{pointScaleId}

Summary

Returns a single pointScale

Description

Returns a single pointScale

Parameters

Name	Located in	Description	Required	Schema
pointScaleId	path	ID of pointScale to fetch	Yes	⇔ integer (int64)
token	header	token to be passed as a header	Yes	⇔ string

Responses

Code	Description	Schema
200	pointScale retrieved	⇔ <div>▼ pointScale { name: string *</div>
403	Invalid token	
404	PointScale doesn't exist	

Try this operation

PUT /pointScales/{pointScaleId}

Summary

Update an existing pointScale

Description

Update an existing pointScale

Parameters

Name	Located in	Description	Required	Schema
pointScale	body	pointScale object to add to the store	Yes	▼ pointScale { ⇒ name: string * }
pointScaleId	path	Id of the pointScale that needs to be updated	Yes	⇒ integer (int64)
token	header	token to be passed as a header	Yes	⇒ string

Responses

Code	Description
200	pointScale updated
403	Invalid token
404	Not found

[Try this operation](#)

DELETE /pointScales/{pointScaleId}

Summary

Delete an existing pointScale

Description

Delete an existing pointScale

Parameters

Name	Located in	Description	Required	Schema
pointScaleId	path	Id of the pointScale that needs to be deleted	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
200	pointScale deleted successfully
403	Invalid token
404	pointScale not found
422	Invalid id supplied

Try this operation

/events

POST /events

Summary

Post events on the platform

Description

Creates a new event

Parameters

Name	Located in	Description	Required	Schema
event	body	event object to add to the platform	Yes	<div><div>▼ event {</div><div>username: string*</div><div>type: string*</div></div> }

Responses

Code	Description
201	event response
403	Invalid token

Try this operation

/application

POST /application

Summary

Post new application on the platform

Description

Creates a new application

Parameters

Name	Located in	Description	Required	Schema
application	body	application object to add to the platform	Yes	<div><div>▼ application {</div><div>name: string *</div><div>password: string *</div><div>}</div></div>

Responses

Code	Description	Schema
201	application response	<div><div>▼ token {</div><div>token: string *</div><div>}</div></div>
400	Invalid payload	
403	Invalid token	
409	Conflict	

Try this operation

DELETE /application

Summary

Delete an existing application

Description

Delete an existing application

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
204	Application deleted successfully
403	Invalid token

Try this operation

/auth

POST /auth

Summary

Returns an application token

Description

Returns the application token

Parameters

Name	Located in	Description	Required	Schema
application	body	application object to get	Yes	\Rightarrow <pre>▼ credentials { name: string * password: string * }</pre>

Responses

Code	Description	Schema
200	Successful authentication	\Rightarrow <pre>▼ token { token: string * }</pre>
400	Application not found	
401	Invalid password	

Try this operation

/users

GET /users

Summary

Returns every users


Description

Returns every users

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	⇒ string

Responses

Code	Description	Schema
200	users response	⇒ 
403	Invalid token	

Try this operation

/leaderboard

GET /leaderboard

Summary

Returns the leaderboard


Description

Returns the leaderboard

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	⇒ string

Responses

Code	Description	Schema
200	leaderboard response	⇒ 
403	Invalid token	

Try this operation

/users/{userName}/badges

GET /users/{userName}/badges

Summary

Returns a user's badges

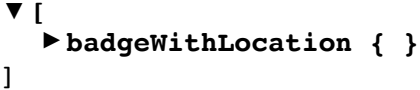
Description

Returns a user's badges

Parameters

Name	Located in	Description	Required	Schema
userName	path	name of the user	Yes	\Rightarrow string
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description	Schema
200	badges response	\Rightarrow  <pre>▼ [► badgeWithLocation { }</pre>
403	Invalid token	

Try this operation

/users/{userName}/pointScales

GET /users/{userName}/pointScales

Summary

Returns a user's pointScales


Description

Returns a user's pointScales

Parameters

Name	Located in	Description	Required	Schema
userName	path	name of the user	Yes	\Leftrightarrow string
token	header	token to be passed as a header	Yes	\Leftrightarrow string

Responses

Code	Description	Schema
200	pointScales response	\Leftrightarrow 
403	Invalid token	

[Try this operation](#)

/badgeRules

GET /badgeRules

Summary

Returns every badge rules

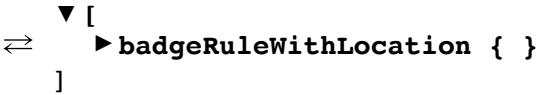
Description

Returns every badge rules

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	\Leftrightarrow string

Responses

Code	Description	Schema
200	badgeRules response	\Leftrightarrow 
403	Invalid token	

[Try this operation](#)

POST /badgeRules

Summary

Creates a new badgeRule

Description

Creates a new badgeRule

Parameters

Name	Located in	Description	Required	Schema
badgeRule	body	badgeRule object to add to the store	Yes	<div><div>▼ badgeRule {</div><div><div>type:</div><div>string*</div></div><div><div>badge:</div><div>integer*</div></div><div><div>pointScale:</div><div>integer*</div></div><div><div>threshold:</div><div>integer*</div></div></div>

Responses

Code	Description	Headers			
		Name	Description	Type	Details
201	badgeRule response	location	Location of the badgeRule	"string"	► Object
403	Invalid token				

Try this operation

/badgeRules/{badgeRuleId}

GET /badgeRules/{badgeRuleId}

Summary

Returns a single badgeRule

Description

Returns a single badgeRule

Parameters

Name	Located in	Description	Required	Schema
badgeRuleId	path	ID of badgeRule to fetch	Yes	⇔ integer (int64)
token	header	token to be passed as a header	Yes	⇔ string

Responses

Code	Description	Schema
200	badgeRule response	⇔ ▼ badgeRule { type: string * badge: integer * pointScale: integer * threshold: integer * }
403	Invalid token	

Try this operation

PUT /badgeRules/{badgeRuleId}

Summary

Update an existing badgeRule

Description

Update an existing badgeRule

Parameters

Name	Located in	Description	Required	Schema
badgeRule	body	badgeRule object to add to the store	Yes	\Rightarrow ▼ badgeRule { type: string * badge: integer * pointScale: integer * threshold: integer * }
badgeRuleId	path	Id of the badgeRule that needs to be updated	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
200	badgeRule response
403	Invalid token'

Try this operation

DELETE /badgeRules/{badgeRuleId}

Summary

Delete an existing badgeRule

Description

Delete an existing badgeRule

Parameters

Name	Located in	Description	Required	Schema
badgeRuleId	path	Id of the badgeRule that needs to be deleted	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
204	badgeRule deleted successfully
400	Invalid Id supplied
404	badgeRule not found

[Try this operation](#)

/pointScaleRules

GET /pointScaleRules

Summary

Returns every pointScale rules

Description

Returns every pointScale rules

Parameters

Name	Located in	Description	Required	Schema
token	header	token to be passed as a header	Yes	⇔ string

Responses

Code	Description	Schema
200	pointScaleRules response	⇔ <div>▼ [▶ pointScaleRuleWithLocation { }]</div>
403	Invalid token	

Try this operation

POST /pointScaleRules

Summary

Creates a new pointScaleRule

Description

Creates a new pointScaleRule

Parameters

Name	Located in	Description	Required	Schema
pointScaleRule	body	pointScaleRule object to add to the store	Yes	<div><div>▼ pointScaleRule {</div><div>type: string *</div><div>pointScale: integer *</div><div>increment: integer *</div><div>}</div></div>
token	header	token to be passed as a header	Yes	⇒ string

Responses

Code	Description	Headers			
201	pointScaleRule response	Name	Description	Type	Details
		location	Location of the pointScaleRule	"string"	► Object
403	Invalid token				

Try this operation

/pointScaleRules/{pointScaleRuleId}

GET /pointScaleRules/{pointScaleRuleId}

Summary

Returns a single pointScaleRule

Description

Returns a single pointScaleRule

Parameters

Name	Located in	Description	Required	Schema
pointScaleRuleId	path	ID of pointScaleRule to fetch	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description	Schema
200	pointScaleRule response	\Rightarrow <pre>▼ pointScaleRule { type: string * pointScale: integer * increment: integer * }</pre>
403	Invalid token	

Try this operation

PUT /pointScaleRules/{pointScaleRuleId}

Summary

Update an existing pointScaleRule

Description

Update an existing pointScaleRule

Parameters

Name	Located in	Description	Required	Schema
pointScaleRule	body	pointScaleRule object to add to the store	Yes	\Rightarrow <pre>▼ pointScaleRule { type: string * pointScale: integer * increment: integer * }</pre>
pointScaleRuleId	path	Id of the pointScaleRule that needs to be updated	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
200	pointScaleRule response
403	Invalid token

Try this operation

DELETE /pointScaleRules/{pointScaleRuleId}

Summary

Delete an existing pointScaleRule

Description

Delete an existing pointScaleRule

Parameters

Name	Located in	Description	Required	Schema
pointScaleRuleId	path	Id of the pointScaleRule that needs to be deleted	Yes	\Rightarrow integer (int64)
token	header	token to be passed as a header	Yes	\Rightarrow string

Responses

Code	Description
204	pointScaleRule deleted successfully
400	Invalid Id supplied
404	pointScaleRule not found

[Try this operation](#)

Models

badge

```
▼ badge {
  ⇒ name: string *
}
```

badgeWithLocation

```
▼ badgeWithLocation {
  ⇒ name: string *
  location: string *
}
```

pointScale

```
▼ pointScale {
  ⇒ name: string *
}
```

pointScaleWithLocation

```
▼ pointScaleWithLocation {  
⇒   name:      string *  
    location: string *  
}
```

userPointScale

```
▼ userPointScale {  
    name:      string *  
⇒   points:   integer *  
    location: string *  
}
```

user

```
▼ user {  
⇒   id:        integer *  
    username: string *  
}
```

userWithNumberOfBadges

```
▼ userWithNumberOfBadges {  
⇒   username:      string *  
    numberOfBadges: integer *  
}
```

event

```
▼ event {  
⇒   username: string *  
    type:     string *  
}
```

badgeRule

```
▼ badgeRule {  
    type:      string *  
⇒   badge:     integer *  
    pointScale: integer *  
    threshold: integer *  
}
```

badgeRuleWithLocation

```
▼ badgeRuleWithLocation {  
    type:      string *  
    badge:     integer *  
⇒   pointScale: integer *  
    threshold: integer *  
    location:  string *  
}
```

pointScaleRule

```
▼ pointScaleRule {  
  type:      string *  
⇒ pointScale: integer *  
  increment: integer *  
}
```

pointScaleRuleWithLocation

```
▼ pointScaleRuleWithLocation {  
  location:  string *  
⇒ type:      string *  
  pointScale: integer *  
  increment: integer *  
}
```

application

```
▼ application {  
⇒ name:      string *  
  password: string *  
}
```

credentials

```
▼ credentials {  
⇒ name:      string *  
  password: string *  
}
```

token

```
▼ token {  
⇒ token: string *  
}
```