

Little Hero API

Table of Contents

1. Overview	1
1.1. Version information	1
1.2. Contact information.....	1
1.3. URI scheme.....	1
1.4. Tags	1
1.5. Consumes	1
1.6. Produces	1
2. Resources	2
2.1. V1/avatar-item-links.....	2
2.1.1. Create avatar	2
2.1.2. Returns list of avatar-item links or filtered list of links given avatar id or item id	2
2.1.3. Returns avatar-item entity	3
2.1.4. Delete a avatar-item entity given its identifier	3
2.1.5. Update avatar_item entity given only its parameters that should be updated	4
2.2. V1/avatars.....	4
2.2.1. Create avatar	4
2.2.2. Returns list of avatars or filtered list of avatars given child id or tutor id	5
2.2.3. Create notifications	5
2.2.4. Returns list of notifications	6
2.2.5. Return notification	6
2.2.6. Delete an notification given its identifier	7
2.2.7. Update status of notification	8
2.2.8. Returns avatar	8
2.2.9. Delete a avatar given its identifier	9
2.2.10. Update Avatar given only its parameters that should be updated	9
2.3. V1/children.....	10
2.3.1. Create child	10
2.3.2. Returns list of children	10
2.3.3. Return invitation	11
2.3.4. Delete an invitation given its identifier.....	11
2.3.5. Update status of invitation	12
2.3.6. Returns child	12
2.3.7. Delete a child given its identifier	13
2.3.8. Update child given only its parameters that should be updated	13
2.3.9. Create invitations	14
2.3.10. Returns list of invitations	14

2.3.11. GET /v1/children/{id}/tutors	15
2.4. V1/documentation	16
2.4.1. GET /v1/documentation/	16
2.5. V1/items	16
2.5.1. Create item	16
2.5.2. Returns list of items	16
2.5.3. Bulk update items	17
2.5.4. Returns item	17
2.5.5. Delete a item given its identifier	18
2.5.6. Update item given only its parameters that should be updated	18
2.6. V1/tasks	19
2.6.1. Create task	19
2.6.2. Returns list of tasks or filtered list of task given avatar id or tutor id	19
2.6.3. Returns task	20
2.6.4. Delete a task given its identifier	20
2.6.5. Update task given only its parameters that should be updated	21
2.7. V1/tutors	21
2.7.1. Create tutor	21
2.7.2. Returns list of tutors	22
2.7.3. Returns tutor	22
2.7.4. Delete a tutor given its identifier	23
2.7.5. Update tutor given only its parameters that should be updated	23
2.7.6. GET /v1/tutors/{id}/children	24
2.7.7. Create invitations	24
2.7.8. Returns list of invitations	25
2.7.9. Return invitation	25
2.7.10. Delete an invitation given its identifier	26
2.7.11. Update status of invitation	26
3. Definitions	28
3.1. Avatar entity	28
3.2. Avatar entity for patch request	28
3.3. Avatar for post	29
3.4. Avatar-Item entity	30
3.5. Avatar-Item entity for patch request	30
3.6. Avatar-Item for post	31
3.7. Child entity	31
3.8. Child entity for patch request	31
3.9. Child for post	32

3.10. Child invitation for post	32
3.11. Errors fields	32
3.12. Full notification	32
3.13. Invitation entity	33
3.14. Invitation for patch	33
3.15. Item entity	33
3.16. Item entity for patch request	34
3.17. Item for post	35
3.18. Notification for patch	35
3.19. Notification for post	35
3.20. Task entity	36
3.21. Task entity for patch request	36
3.22. Task for post	36
3.23. Tutor entity	37
3.24. Tutor entity for patch request	37
3.25. Tutor for post	37
3.26. Tutor invitation for post	38

Chapter 1. Overview

API for Little Hero mobile application

1.1. Version information

Version : 1.2

1.2. Contact information

Contact : Dawid Jurkiewicz

Contact Email : dawjur@st.amu.edu.pl

1.3. URI scheme

BasePath : /api

1.4. Tags

- default : Default namespace
- v1/children : Operations related to children
- v1/avatars : Operations related to avatars
- v1/tutors : Operations related to tutors
- v1/items : Operations related to items
- v1/tasks : Operations related to tasks
- v1/avatar-item-links : Operations related to avatar item links
- v1/documentation : Endpoint for getting documentation in json

1.5. Consumes

- `application/json`

1.6. Produces

- `application/json`

Unresolved directive in index.adoc - include::manual_content1.adoc[]

Unresolved directive in index.adoc - include::manual_content2.adoc[]

Chapter 2. Resources

2.1. V1/avatar-item-links

Operations related to avatar item links

2.1.1. Create avatar

```
POST /v1/avatar-item-links/
```

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Avatar-Item for post

Responses

HTTP Code	Description	Schema
200	Success	Avatar-Item entity
201	Avatar-Item link created.	No Content
400	Bad request	No Content

2.1.2. Returns list of avatar-item links or filtered list of links given avatar id or item id

```
GET /v1/avatar-item-links/
```

Parameters

Type	Name	Description	Schema
Query	avatar_id <i>optional</i>	For filtering by avatar id	string
Query	item_id <i>optional</i>	For filtering by item id	string
Query	state <i>optional</i>	For filtering by item state	string

Responses

HTTP Code	Description	Schema
200	Success	< Avatar-Item entity > array
400	Bad request	No Content

2.1.3. Returns avatar-item entity

```
GET /v1/avatar-item-links/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Avatar-Item entity
400	Bad request	No Content
404	Avatar-Item link not found	No Content

2.1.4. Delete a avatar-item entity given its identifier

```
DELETE /v1/avatar-item-links/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer

Responses

HTTP Code	Description	Schema
204	Avatar-Item deleted	No Content
400	Bad request	No Content
404	Avatar-Item link not found	No Content

2.1.5. Update avatar_item entity given only its parameters that should be updated

```
PATCH /v1/avatar-item-links/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer
Body	payload <i>required</i>		Avatar-Item entity for patch request

Responses

HTTP Code	Description	Schema
200	Avatar-Item updated	No Content
400	Bad request	No Content
404	Avatar-Item link not found	No Content

2.2. V1/avatars

Operations related to avatars

2.2.1. Create avatar

```
POST /v1/avatars/
```

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Avatar for post

Responses

HTTP Code	Description	Schema
200	Success	Avatar entity
201	Avatar created.	No Content
400	Bad request	No Content

2.2.2. Returns list of avatars or filtered list of avatars given child id or tutor id

```
GET /v1/avatars/
```

Parameters

Type	Name	Description	Schema
Query	child_id <i>optional</i>	For filtering by child id	string
Query	tutor_id <i>optional</i>	For filtering by tutor id	string

Responses

HTTP Code	Description	Schema
200	Success	< Avatar entity > array
400	Bad request	No Content

2.2.3. Create notifications

```
POST /v1/avatars/{avatar_id}/notifications
```

Parameters

Type	Name	Description	Schema
Path	avatar_id <i>required</i>	The avatar identifier	integer
Body	payload <i>required</i>		Notification for post

Responses

HTTP Code	Description	Schema
200	Success	Full notification
201	Notification created.	No Content
400	Bad request	No Content
404	Avatar not found	No Content

2.2.4. Returns list of notifications

```
GET /v1/avatars/{avatar_id}/notifications
```

Parameters

Type	Name	Description	Schema
Path	avatar_id <i>required</i>	The avatar identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	< Full notification > array
400	Bad request	No Content
404	Avatar not found	No Content

2.2.5. Return notification

```
GET /v1/avatars/{avatar_id}/notifications/{notification_id}
```

Parameters

Type	Name	Description	Schema
Path	avatar_id <i>required</i>	The avatar identifier	integer
Path	notification_id <i>required</i>	The notification identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Full notification
400	Bad request	No Content
404	Avatar or notification not found	No Content

2.2.6. Delete an notification given its identifier

```
DELETE /v1/avatars/{avatar_id}/notifications/{notification_id}
```

Parameters

Type	Name	Description	Schema
Path	avatar_id <i>required</i>	The avatar identifier	integer
Path	notification_id <i>required</i>	The notification identifier	integer

Responses

HTTP Code	Description	Schema
204	Notification deleted	No Content
400	Bad request	No Content
404	Avatar or notification not found	No Content

2.2.7. Update status of notification

```
PATCH /v1/avatars/{avatar_id}/notifications/{notification_id}
```

Parameters

Type	Name	Description	Schema
Path	avatar_id <i>required</i>	The avatar identifier	integer
Path	notification_id <i>required</i>	The notification identifier	integer
Body	payload <i>required</i>		Notification for patch

Responses

HTTP Code	Description	Schema
200	Notification updated	No Content
400	Bad request	No Content
404	Avatar or notification not found	No Content

2.2.8. Returns avatar

```
GET /v1/avatars/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Avatar entity

HTTP Code	Description	Schema
400	Bad request	No Content
404	Avatar not found	No Content

2.2.9. Delete a avatar given its identifier

```
DELETE /v1/avatars/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer

Responses

HTTP Code	Description	Schema
204	Avatar deleted	No Content
400	Bad request	No Content
404	Avatar not found	No Content

2.2.10. Update Avatar given only its parameters that should be updated

```
PATCH /v1/avatars/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The avatar identifier	integer
Body	payload <i>required</i>		Avatar entity for patch request

Responses

HTTP Code	Description	Schema
200	Avatar updated	No Content
400	Bad request	No Content
404	Avatar not found	No Content

2.3. V1/children

Operations related to children

2.3.1. Create child

POST /v1/children/

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Child for post

Responses

HTTP Code	Description	Schema
200	Success	Child entity
201	Child created.	No Content
400	Bad request	No Content

2.3.2. Returns list of children

GET /v1/children/

Responses

HTTP Code	Description	Schema
200	Success	< Child entity > array

HTTP Code	Description	Schema
400	Bad request	No Content

2.3.3. Return invitation

```
GET /v1/children/{child_id}/invitations/{invitation_id}
```

Parameters

Type	Name	Description	Schema
Path	child_id <i>required</i>	The child identifier	integer
Path	invitation_id <i>required</i>	The invitation identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Invitation entity
400	Bad request	No Content
404	Child not found	No Content

2.3.4. Delete an invitation given its identifier

```
DELETE /v1/children/{child_id}/invitations/{invitation_id}
```

Parameters

Type	Name	Description	Schema
Path	child_id <i>required</i>	The child identifier	integer
Path	invitation_id <i>required</i>	The invitation identifier	integer

Responses

HTTP Code	Description	Schema
204	Invitation deleted	No Content
400	Bad request	No Content
404	Child not found	No Content

2.3.5. Update status of invitation

```
PATCH /v1/children/{child_id}/invitations/{invitation_id}
```

Parameters

Type	Name	Description	Schema
Path	child_id <i>required</i>	The child identifier	integer
Path	invitation_id <i>required</i>	The invitation identifier	integer
Body	payload <i>required</i>		Invitation for patch

Responses

HTTP Code	Description	Schema
200	Invitation updated	No Content
400	Bad request	No Content
404	Child not found	No Content

2.3.6. Returns child

```
GET /v1/children/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Child entity
400	Bad request	No Content
404	Child not found	No Content

Security

Type	Name
Unknown	Basic

2.3.7. Delete a child given its identifier

```
DELETE /v1/children/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer

Responses

HTTP Code	Description	Schema
204	Child deleted	No Content
400	Bad request	No Content
404	Child not found	No Content

2.3.8. Update child given only its parameters that should be updated

```
PATCH /v1/children/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer
Body	payload <i>required</i>		Child entity for patch request

Responses

HTTP Code	Description	Schema
200	Child updated	No Content
400	Bad request	No Content
404	Child not found	No Content

2.3.9. Create invitations

POST /v1/children/{id}/invitations

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer
Body	payload <i>required</i>		Child invitation for post

Responses

HTTP Code	Description	Schema
200	Success	Invitation entity
201	Invitation created.	No Content
400	Bad request	No Content
404	Child not found	No Content

2.3.10. Returns list of invitations

GET /v1/children/{id}/invitations

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer
Query	kind <i>optional</i>	For filtering by type of invitation	string
Query	status <i>optional</i>	For filtering by status of invitation	string

Responses

HTTP Code	Description	Schema
200	Success	< Invitation entity > array
400	Bad request	No Content
404	Child not found	No Content

2.3.11. GET /v1/children/{id}/tutors

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The child identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	< Tutor entity > array
400	Bad request	No Content
404	Child not found	No Content

2.4. V1/documentation

Endpoint for getting documentation in json

2.4.1. GET /v1/documentation/

Responses

HTTP Code	Description	Schema
200	Success	No Content

2.5. V1/items

Operations related to items

2.5.1. Create item

POST /v1/items/

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Item for post

Responses

HTTP Code	Description	Schema
200	Success	Item entity
201	Item created.	No Content
400	Bad request	No Content

2.5.2. Returns list of items

GET /v1/items/

Responses

HTTP Code	Description	Schema
200	Success	< Item entity > array
400	Bad request	No Content

2.5.3. Bulk update items

```
PUT /v1/items/
```

Parameters

Type	Name	Schema
Body	payload <i>required</i>	< Item for post > array

Responses

HTTP Code	Description	Schema
204	No content	No Content
400	Bad request	No Content

2.5.4. Returns item

```
GET /v1/items/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The item identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Item entity
400	Bad request	No Content
404	Item not found	No Content

2.5.5. Delete a item given its identifier

```
DELETE /v1/items/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The item identifier	integer

Responses

HTTP Code	Description	Schema
204	item deleted	No Content
400	Bad request	No Content
404	Item not found	No Content

2.5.6. Update item given only its parameters that should be updated

```
PATCH /v1/items/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The item identifier	integer
Body	payload <i>required</i>		Item entity for patch request

Responses

HTTP Code	Description	Schema
200	item updated	No Content
400	Bad request	No Content
404	Item not found	No Content

2.6. V1/tasks

Operations related to tasks

2.6.1. Create task

POST /v1/tasks/

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Task for post

Responses

HTTP Code	Description	Schema
200	Success	Task entity
201	task created.	No Content
400	Bad request	No Content

2.6.2. Returns list of tasks or filtered list of task given avatar id or tutor id

GET /v1/tasks/

Parameters

Type	Name	Description	Schema
Query	avatar_id <i>optional</i>	For filtering by avatar id	string
Query	tutor_id <i>optional</i>	For filtering by tutor id	string

Responses

HTTP Code	Description	Schema
200	Success	< Task entity > array
400	Bad request	No Content

2.6.3. Returns task

```
GET /v1/tasks/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The task identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Task entity
400	Bad request	No Content
404	Task not found	No Content

2.6.4. Delete a task given its identifier

```
DELETE /v1/tasks/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The task identifier	integer

Responses

HTTP Code	Description	Schema
204	Task deleted	No Content
400	Bad request	No Content
404	Task not found	No Content

2.6.5. Update task given only its parameters that should be updated

```
PATCH /v1/tasks/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The task identifier	integer
Body	payload <i>required</i>		Task entity for patch request

Responses

HTTP Code	Description	Schema
200	Task updated	No Content
400	Bad request	No Content
404	Task not found	No Content

2.7. V1/tutors

Operations related to tutors

2.7.1. Create tutor

POST /v1/tutors/

Parameters

Type	Name	Schema
Body	payload <i>required</i>	Tutor for post

Responses

HTTP Code	Description	Schema
200	Success	Tutor entity
201	Tutor created.	No Content
400	Bad request	No Content

2.7.2. Returns list of tutors

GET /v1/tutors/

Responses

HTTP Code	Description	Schema
200	Success	< Tutor entity > array
400	Bad request	No Content

2.7.3. Returns tutor

GET /v1/tutors/{id}

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Tutor entity
400	Bad request	No Content
404	Tutor not found	No Content

2.7.4. Delete a tutor given its identifier

```
DELETE /v1/tutors/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer

Responses

HTTP Code	Description	Schema
204	Tutor deleted	No Content
400	Bad request	No Content
404	Tutor not found	No Content

2.7.5. Update tutor given only its parameters that should be updated

```
PATCH /v1/tutors/{id}
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer
Body	payload <i>required</i>		Tutor entity for patch request

Responses

HTTP Code	Description	Schema
200	Success	Tutor entity
201	Tutor updated	No Content
400	Bad request	No Content
404	Tutor not found	No Content

2.7.6. GET /v1/tutors/{id}/children

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	< Child entity > array
400	Bad request	No Content
404	Tutor not found	No Content

2.7.7. Create invitations

```
POST /v1/tutors/{id}/invitations
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer
Body	payload <i>required</i>		Tutor invitation for post

Responses

HTTP Code	Description	Schema
200	Success	Invitation entity
201	Invitation created.	No Content
400	Bad request	No Content
404	Tutor not found	No Content

2.7.8. Returns list of invitations

```
GET /v1/tutors/{id}/invitations
```

Parameters

Type	Name	Description	Schema
Path	id <i>required</i>	The tutor identifier	integer
Query	kind <i>optional</i>	For filtering by type of invitation	string
Query	status <i>optional</i>	For filtering by status of invitation	string

Responses

HTTP Code	Description	Schema
200	Success	< Invitation entity > array
400	Bad request	No Content
404	Tutor not found	No Content

2.7.9. Return invitation

```
GET /v1/tutors/{tutor_id}/invitations/{invitation_id}
```

Parameters

Type	Name	Description	Schema
Path	invitation_id <i>required</i>	The invitation identifier	integer
Path	tutor_id <i>required</i>	The tutor identifier	integer

Responses

HTTP Code	Description	Schema
200	Success	Invitation entity
400	Bad request	No Content
404	Tutor not found	No Content

2.7.10. Delete an invitation given its identifier

```
DELETE /v1/tutors/{tutor_id}/invitations/{invitation_id}
```

Parameters

Type	Name	Description	Schema
Path	invitation_id <i>required</i>	The invitation identifier	integer
Path	tutor_id <i>required</i>	The tutor identifier	integer

Responses

HTTP Code	Description	Schema
204	Invitation deleted	No Content
400	Bad request	No Content
404	Tutor not found	No Content

2.7.11. Update status of invitation

PATCH /v1/tutors/{tutor_id}/invitations/{invitation_id}

Parameters

Type	Name	Description	Schema
Path	invitation_id <i>required</i>	The invitation identifier	integer
Path	tutor_id <i>required</i>	The tutor identifier	integer
Body	payload <i>required</i>		Invitation for patch

Responses

HTTP Code	Description	Schema
200	Invitation updated	No Content
400	Bad request	No Content
404	Tutor not found	No Content

Chapter 3. Definitions

3.1. Avatar entity

Name	Description	Schema
child_id <i>required</i>	Id of avatars owner (child)	integer
clazz <i>required</i>	Avatar class Example : "human"	enum (human, king, mage, cowboy)
experience <i>optional</i>	Avatar experience points Minimum value : 0 Maximum value : 100	integer
health <i>required</i>	Avatar health Minimum value : 0 Maximum value : 100	integer
id <i>optional</i>	Id of avatar	integer
level <i>required</i>	Avatar level Minimum value : 0	integer
money <i>required</i>	Avatar money Minimum value : 0	integer
name <i>required</i>	Avatar name	string
tutor_id <i>required</i>	Id of tutors related to avatar	integer
update_invitation <i>required</i>	Flag indicating if invitations changed	boolean
update_notification <i>required</i>	Flag indicating if notification is required	boolean
update_task <i>required</i>	Flag indicating if update of task is required	boolean

3.2. Avatar entity for patch request

Name	Description	Schema
child_id <i>optional</i>	Id of avatars owner (child)	integer
clazz <i>optional</i>	Avatar class Example : "human"	enum (human, king, mage, cowboy)
experience <i>optional</i>	Avatar experience points Minimum value : 0 Maximum value : 100	integer
health <i>optional</i>	Avatar health Minimum value : 0 Maximum value : 100	integer
level <i>optional</i>	Avatar level Minimum value : 0	integer
money <i>optional</i>	Avatar money Minimum value : 0	integer
name <i>optional</i>	Avatar name	string
tutor_id <i>optional</i>	Id of tutors related to avatar	integer
update_invitation <i>optional</i>	Flag indicating if invitations changed	boolean
update_notification <i>optional</i>	Flag indicating if notification is required	boolean
update_task <i>optional</i>	Flag indicating if update of task is required	boolean

3.3. Avatar for post

Name	Description	Schema
child_id <i>required</i>	Id of avatars owner (child)	integer
clazz <i>required</i>	Avatar class Example : "human"	enum (human, king, mage, cowboy)
experience <i>optional</i>	Avatar experience points Minimum value : 0 Maximum value : 100	integer

Name	Description	Schema
health <i>required</i>	Avatar health Minimum value : 0 Maximum value : 100	integer
level <i>required</i>	Avatar level Minimum value : 0	integer
money <i>required</i>	Avatar money Minimum value : 0	integer
name <i>required</i>	Avatar name	string
tutor_id <i>required</i>	Id of tutors related to avatar	integer

3.4. Avatar-Item entity

Name	Description	Schema
avatar_id <i>required</i>	Avatar id	integer
id <i>optional</i>	Id of avatar-item	integer
item_id <i>required</i>	Item id	integer
state <i>required</i>	State of item (on, bought, available, unavailable) Example : "worn"	enum (worn, canBePutOn, canBePurchased, unavailable)

3.5. Avatar-Item entity for patch request

Name	Description	Schema
avatar_id <i>optional</i>	Avatar id	integer
item_id <i>optional</i>	Item id	integer
state <i>optional</i>	State of item (worn, canBePutOn, canBePurchased, unavailable) Example : "worn"	enum (worn, canBePutOn, canBePurchased, unavailable)

3.6. Avatar-Item for post

Name	Description	Schema
avatar_id <i>required</i>	Avatar id	integer
item_id <i>required</i>	Item id	integer
state <i>required</i>	State of item (on, bought, available, unavailable) Example : "worn"	enum (worn, canBePutOn, canBePurchased, unavailable)

3.7. Child entity

Name	Description	Schema
id <i>optional</i>	Id of child	integer
login <i>required</i>	Child login	string
mail <i>required</i>	Child e-mail Pattern : "(^@[a-zA-Z0-9-]\.\.[a-zA-Z0-9-]+\.)"	string
nickname <i>required</i>	Child nickname	string
password <i>required</i>	Child password Minimum length : 12	string

3.8. Child entity for patch request

Name	Description	Schema
login <i>optional</i>	Child login	string
mail <i>optional</i>	Child e-mail Pattern : "(^@[a-zA-Z0-9-]\.\.[a-zA-Z0-9-]+\.)"	string
nickname <i>optional</i>	Child nickname	string
password <i>optional</i>	Child password Minimum length : 12	string

3.9. Child for post

Name	Description	Schema
login <i>required</i>	Child login	string
mail <i>required</i>	Child e-mail Pattern : "(^[a-zA-Z0-9-]\.\.[a-zA-Z0-9-]+\.)"	string
nickname <i>required</i>	Child nickname	string
password <i>required</i>	Child password Minimum length : 12	string

3.10. Child invitation for post

Name	Description	Schema
kind <i>required</i>	Type of invitation Example : "child-avatar"	enum (child-avatar, child, tutor, tutor-avatar)
tutor_id <i>required</i>	Tutor id	integer

3.11. Errors fields

Name	Description	Schema
code <i>required</i>	HTTP status code	integer
message <i>required</i>	Error message	string

3.12. Full notification

Name	Description	Schema
avatar_id <i>required</i>	Id of avatar who has that notification	integer
content <i>required</i>	Content of notification	string

Name	Description	Schema
id <i>optional</i>	Id of notification	integer

3.13. Invitation entity

Name	Description	Schema
child_id <i>required</i>	Child id	integer
id <i>optional</i>	Id of item	integer
kind <i>required</i>	Type of invitation Example : "child-avatar"	enum (child-avatar, child, tutor, tutor-avatar)
status <i>required</i>	Status of invitation Example : "pending"	enum (pending, accepted)
tutor_id <i>required</i>	Tutor id	integer

3.14. Invitation for patch

Name	Description	Schema
status <i>optional</i>	Status of invitation Example : "pending"	enum (pending, accepted)

3.15. Item entity

Name	Description	Schema
clazz <i>required</i>	Item class Example : "wornByDefault"	enum (wornByDefault, allclazzes, human, empty, king, mage, cowboy)
iconSrc <i>required</i>	Path to item icon	string
id <i>optional</i>	Id of item	integer

Name	Description	Schema
imgSrc <i>required</i>	Path to item image	string
level <i>required</i>	Item level. Item is available from this level. Minimum value : 0	integer
price <i>required</i>	Item price Minimum value : 0	integer
type <i>required</i>	Item type Example : "shoes"	enum (shoes, sock, prop_right, prop_left, misc_neck, misc_ear, misc_head, top_bottom, bottom, top, animal, hair_back, hair_front, mouth, nose, eyes)

3.16. Item entity for patch request

Name	Description	Schema
clazz <i>optional</i>	Item class Example : "wornByDefault"	enum (wornByDefault, allclazzes, human, empty, king, mage, cowboy)
iconSrc <i>optional</i>	Path to item icon	string
imgSrc <i>optional</i>	Path to item image	string
level <i>optional</i>	Item level. Item is available from this level. Minimum value : 0	integer
price <i>optional</i>	Item price Minimum value : 0	integer
type <i>optional</i>	Item type Example : "shoes"	enum (shoes, sock, prop_right, prop_left, misc_neck, misc_ear, misc_head, top_bottom, bottom, top, animal, hair_back, hair_front, mouth, nose, eyes)

3.17. Item for post

Name	Description	Schema
clazz <i>required</i>	Item class Example : "wornByDefault"	enum (wornByDefault, allclazzes, human, empty, king, mage, cowboy)
iconSrc <i>required</i>	Path to item icon	string
imgSrc <i>required</i>	Path to item image	string
level <i>required</i>	Item level. Item is available from this level. Minimum value : 0	integer
price <i>required</i>	Item price Minimum value : 0	integer
type <i>required</i>	Item type Example : "shoes"	enum (shoes, sock, prop_right, prop_left, misc_neck, misc_ear, misc_head, top_bottom, bottom, top, animal, hair_back, hair_front, mouth, nose, eyes)

3.18. Notification for patch

Name	Description	Schema
avatar_id <i>optional</i>	Id of avatar who has that notification	integer
content <i>optional</i>	Content of notification	string

3.19. Notification for post

Name	Description	Schema
content <i>required</i>	Content of notification	string

3.20. Task entity

Name	Description	Schema
avatar_id <i>required</i>	Id of avatar who was given this task	integer
content <i>required</i>	Description of task	string
experience <i>required</i>	Experience to be gained after completing task Minimum value : 0 Maximum value : 100	integer
id <i>optional</i>	Id of item	integer
reward <i>required</i>	Money reward after completing task Minimum value : 0	integer
tutor_id <i>required</i>	Id of tutor who was gave the task	integer

3.21. Task entity for patch request

Name	Description	Schema
avatar_id <i>optional</i>	Id of avatar who was given this task	integer
content <i>optional</i>	Description of task	string
experience <i>optional</i>	Experience to be gained after completing task Minimum value : 0 Maximum value : 100	integer
reward <i>optional</i>	Money reward after completing task Minimum value : 0	integer
tutor_id <i>optional</i>	Id of tutor who was gave the task	integer

3.22. Task for post

Name	Description	Schema
avatar_id <i>required</i>	Id of avatar who was given this task	integer

Name	Description	Schema
content <i>required</i>	Description of task	string
experience <i>required</i>	Experience to be gained after completing task Minimum value : 0 Maximum value : 100	integer
reward <i>required</i>	Money reward after completing task Minimum value : 0	integer
tutor_id <i>required</i>	Id of tutor who was gave the task	integer

3.23. Tutor entity

Name	Description	Schema
id <i>optional</i>	Id of tutor	integer
login <i>optional</i>	Tutor login	string
mail <i>required</i>	Tutor e-mail Pattern : "(^@[a-zA-Z0-9-]\.\.[a-zA-Z0-9-]+\.)"	string
password <i>required</i>	Tutor password Minimum length : 12	string

3.24. Tutor entity for patch request

Name	Description	Schema
mail <i>optional</i>	Tutor e-mail Pattern : "(^@[a-zA-Z0-9-]\.\.[a-zA-Z0-9-]+\.)"	string
password <i>optional</i>	Tutor password Minimum length : 12	string

3.25. Tutor for post

Name	Description	Schema
login <i>optional</i>	Tutor login	string

Name	Description	Schema
mail <i>required</i>	Tutor e-mail Pattern : "(^@[a-zA-Z0-9-]\\.[a-zA-Z0-9-\\.]+)\$"	string
password <i>required</i>	Tutor password Minimum length : 12	string

3.26. Tutor invitation for post

Name	Description	Schema
child_id <i>required</i>	Child id	integer
kind <i>required</i>	Type of invitation Example : "child-avatar"	enum (child-avatar, child, tutor, tutor-avatar)