
preguntale Documentation

Release 0.1

Milton Mazzarri

March 12, 2014

Contents:

Questions

- *Get questions*
- *Get a question by ID*
- *Get supporters of a question*
- *Create a question*

Note:

- Add support for [HATEOAS](#)
 - Show the available fields for every question
-

GET /questions

Get the details about the last questions by default.

Example request:

```
GET /questions?fields="question,tags"&limit=10&sort="desc"
Host: example.com
Accept: application/vnd.preguntale.v1+json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
[
  {
    "_id": 1234,
    "question": "Question 1",
    "tags": ["tag1", "tag2"]
  },
  {
    "_id": 1235,
    "question": "Question 2",
    "tags": ["tag1", "tag2"]
  }
]
```

Query Parameters

- **question_ids** (*str*) – A list of comma separated question ids.

- **fields** (*str*) – A list of comma separated question fields. Otherwise all the available fields should be returned.
- **per_page** (*int*) – The maximum number of questions objects that will be returned by request, no more than 100.
- **page** (*int*) – The offset by *per_page* questions. Otherwise, returns the first page of questions.
- **sort** (*str*) – The order by which the questions will be returned based on the date and time the question was created, *asc* or *desc*. Default: *desc*.
- **to** (*int*) – The ID of the person who's acting as a receiver of a question.
- **from** (*int*) – The ID of the person who's acting as sender of a question.

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day
- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **200** – no error
- **400** – bad request
- **404** – there's no questions

GET `/questions/` (**int**: *question_id*)

Get the details about a question based on the ID.

Example request:

```
GET /questions/1234?fields="question,tags"
Host example.com
Accept: application/vnd.preguntale.v1+json
```

Example response:

```
{
  "_id": 1234,
  "question": "Question 1",
  "tags": ["tag1", "tag2"]
}
```

Parameters

- **question_id** (*int*) – ID of the question

Query Parameters

- **fields** (*str*) – A list of comma separated question fields. Otherwise all the available fields should be returned.

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day
- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **200** – no error
- **400** – bad request
- **404** – There is no question with this ID

GET /questions/ (*int: question_id*) /supporters

Get the details about the supporters of a question.

Example request:

```
GET /questions/1234/supporters?fields="first_name,last_name,city,country,signed_on"
Host example.com
Accept: application/vnd.preguntale.v1+json
```

Example response:

```
[
  {
    "first_name": "John",
    "last_name": "Doe",
    "city": "Monterrey",
    "country": "mx",
    "signed_on": "date"
  },
  {
    "first_name": "John",
    "last_name": "Doe",
    "city": "Monterrey",
    "country": "mx",
    "signed_on": "date"
  }
]
```

Parameters

- **question_id** (*int*) – ID of the question

Query Parameters

- **fields** (*str*) – A list of comma separated supporters fields. Otherwise all the available fields should be returned.
- **per_page** (*int*) – The maximum number of supporters objects that will be returned by request, no more than 100.
- **page** (*int*) – The offset by *per_page* supporters. Otherwise, returns the first page of supporters.
- **sort** (*str*) – The order by which the list of supporters will be returned based on the date and time the signature was created, *asc* or *desc*. Default: *desc*.

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day
- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **200** – no error
- **400** – bad request
- **404** – There is no question with this ID

POST /questions

Create a new question

Example request:

```
POST /questions
Host example.com
Accept: application/vnd.preguntale.v1+json
Content-Type: application/x-www-form-urlencoded or application/json
```

Example response:

```
{
  "_id": 1234,
  "question": "Question 1",
  "tags": ["tag1", "tag2"]
}
```

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day
- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **201** – created
- **400** – bad request

Users

- *Get users*
- *Get a user by ID*
- *Get questions created by user ID*
- *Get votes by user ID*

Note:

- Add support for [HATEOAS](#)
 - Show the available fields for every user
 - Add support for create a new user
-

GET /users

Get the details about the lasts created users.

Example request:

```
GET /users?fields="first_name,last_name,city,country,signed_on"  
Host example.com  
Accept: application/vnd.preguntale.v1+json
```

Example response:

```
[  
  {  
    "first_name": "John",  
    "last_name": "Doe",  
    "city": "Monterrey",  
    "country": "mx",  
    "signed_on": "date"  
  },  
  {  
    "first_name": "John",  
    "last_name": "Doe",  
    "city": "Monterrey",  
    "country": "mx",  
    "signed_on": "date"  
  }  
]
```

Query Parameters

- **fields** (*str*) – A list of comma separated users fields. Otherwise all the available fields should be returned.
- **per_page** (*int*) – The maximum number of users objects that will be returned by request, no more than 100.
- **page** (*int*) – The offset by *per_page* users. Otherwise, returns the first page of users.
- **sort** (*str*) – The order by which the list of users will be returned based on the date and time the user was created, *asc* or *desc*. Default: *desc*.

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day
- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **200** – no error
- **400** – bad request
- **404** – Nobody has not registered yet.

GET `/users/` (**int:** *user_id*)

Get the details about the user.

Example request:

```
GET /users/1?fields="first_name,last_name,city,country,signed_on"  
Host example.com  
Accept: application/vnd.preguntale.v1+json
```

Example response:

```
{  
  "first_name": "John",  
  "last_name": "Doe",  
  "city": "Monterrey",  
  "country": "mx",  
  "signed_on": "date"  
}
```

Query Parameters

- **fields** (*str*) – A list of comma separated users fields. Otherwise all the available fields should be returned.

Request Headers

- **Accept** – *application/vnd.preguntale.v1+json*

Response Headers

- **Content-Type** – *application/json*
- **X-RateLimit-Limit** – 1000 per day

- **X-RateLimit-Remaining** – Number of requests remaining
- **X-RateLimit-Reset** – The time at which the current rate limit window resets (epoch time)

Status Codes

- **200** – no error
- **400** – bad request
- **404** – There is no user with this ID.

Indices and tables

- *genindex*
- *modindex*
- *search*

/questions

GET /questions, ??

POST /questions, ??

GET /questions/(int:question_id), ??

GET /questions/(int:question_id)/supporters,
??

/users

GET /users, ??

GET /users/(int:user_id), ??