

RealWorld API Spec

This spec was taken from realworld project.

Authentication Header:

Authorization: Token jwt.token.here

JSON Objects returned by API:

Users (for authentication)

```
{
  "user": {
    "email": "jake@jake.jake",
    "token": "jwt.token.here",
    "username": "jake",
    "bio": "I work at statefarm",
    "image": null
  }
}
```

Profile

```
{
  "profile": {
    "username": "jake",
    "bio": "I work at statefarm",
    "image": "https://static.productionready.io/images/smiley-cyrus.jpg",
    "following": false
  }
}
```

Single Article

```
{
  "article": {
    "slug": "how-to-train-your-dragon",
    "title": "How to train your dragon",
    "description": "Ever wonder how?",
    "body": "It takes a Jacobian",
    "tagList": ["dragons", "training"],
  }
}
```

```

    "createdAt": "2016-02-18T03:22:56.637Z",
    "updatedAt": "2016-02-18T03:48:35.824Z",
    "favorited": false,
    "favoritesCount": 0,
    "author": {
      "username": "jake",
      "bio": "I work at statefarm",
      "image": "https://i.stack.imgur.com/xHWG8.jpg",
      "following": false
    }
  }
}

```

Multiple Articles

```

{
  "articles": [{
    "slug": "how-to-train-your-dragon",
    "title": "How to train your dragon",
    "description": "Ever wonder how?",
    "body": "It takes a Jacobian",
    "tagList": ["dragons", "training"],
    "createdAt": "2016-02-18T03:22:56.637Z",
    "updatedAt": "2016-02-18T03:48:35.824Z",
    "favorited": false,
    "favoritesCount": 0,
    "author": {
      "username": "jake",
      "bio": "I work at statefarm",
      "image": "https://i.stack.imgur.com/xHWG8.jpg",
      "following": false
    }
  }], {

    "slug": "how-to-train-your-dragon-2",
    "title": "How to train your dragon 2",
    "description": "So toothless",
    "body": "It a dragon",
    "tagList": ["dragons", "training"],
    "createdAt": "2016-02-18T03:22:56.637Z",
    "updatedAt": "2016-02-18T03:48:35.824Z",
    "favorited": false,
    "favoritesCount": 0,
    "author": {
      "username": "jake",

```

```

        "bio": "I work at statefarm",
        "image": "https://i.stack.imgur.com/xHWG8.jpg",
        "following": false
    }
  ]],
  "articlesCount": 2
}

```

Single Comment

```

{
  "comment": {
    "id": 1,
    "createdAt": "2016-02-18T03:22:56.637Z",
    "updatedAt": "2016-02-18T03:22:56.637Z",
    "body": "It takes a Jacobian",
    "author": {
      "username": "jake",
      "bio": "I work at statefarm",
      "image": "https://i.stack.imgur.com/xHWG8.jpg",
      "following": false
    }
  }
}

```

Multiple Comments

```

{
  "comments": [{
    "id": 1,
    "createdAt": "2016-02-18T03:22:56.637Z",
    "updatedAt": "2016-02-18T03:22:56.637Z",
    "body": "It takes a Jacobian",
    "author": {
      "username": "jake",
      "bio": "I work at statefarm",
      "image": "https://i.stack.imgur.com/xHWG8.jpg",
      "following": false
    }
  }]
}

```

List of Tags

```
{
  "tags": [
    "reactjs",
    "angularjs"
  ]
}
```

Errors and Status Codes

If a request fails any validations, expect a 422 and errors in the following format:

```
{
  "errors":{
    "body": [
      "can't be empty"
    ]
  }
}
```

Other status codes:

401 for Unauthorized requests, when a request requires authentication but it isn't provided

403 for Forbidden requests, when a request may be valid but the user doesn't have permissions to perform the action

404 for Not found requests, when a resource can't be found to fulfill the request

Endpoints:

Authentication:

POST /api/users/login

Example request body:

```
{
  "user":{
    "email": "jake@jake.jake",
    "password": "jakejake"
  }
}
```

No authentication required, returns a User

Required fields: `email`, `password`

Registration:

POST `/api/users`

Example request body:

```
{
  "user":{
    "username": "Jacob",
    "email": "jake@jake.jake",
    "password": "jakejake"
  }
}
```

No authentication required, returns a User

Required fields: `email`, `username`, `password`

Get Current User

GET `/api/user`

Authentication required, returns a User that's the current user

Update User

PUT `/api/user`

Example request body:

```
{
  "user":{
    "email": "jake@jake.jake",
    "bio": "I like to skateboard",
    "image": "https://i.stack.imgur.com/xHWG8.jpg"
  }
}
```

Authentication required, returns the User

Accepted fields: `email`, `username`, `password`, `image`, `bio`

Get Profile

GET /api/profiles/:username

Authentication optional, returns a Profile

Follow user

POST /api/profiles/:username/follow

Authentication required, returns a Profile

No additional parameters required

Unfollow user

DELETE /api/profiles/:username/follow

Authentication required, returns a Profile

No additional parameters required

List Articles

GET /api/articles

Returns most recent articles globally by default, provide **tag**, **author** or **favorited** query parameter to filter results

Query Parameters:

Filter by tag:

?tag=AngularJS

Filter by author:

?author=jake

Favorited by user:

?favorited=jake

Limit number of articles (default is 20):

?limit=20

Offset/skip number of articles (default is 0):

?offset=0

Authentication optional, will return multiple articles, ordered by most recent first

Feed Articles

GET /api/articles/feed

Can also take `limit` and `offset` query parameters like List Articles

Authentication required, will return multiple articles created by followed users, ordered by most recent first.

Get Article

GET /api/articles/:slug

No authentication required, will return single article

Create Article

POST /api/articles

Example request body:

```
{
  "article": {
    "title": "How to train your dragon",
    "description": "Ever wonder how?",
    "body": "You have to believe",
    "tagList": ["reactjs", "angularjs", "dragons"]
  }
}
```

Authentication required, will return an Article

Required fields: `title`, `description`, `body`

Optional fields: `tagList` as an array of Strings

Update Article

PUT /api/articles/:slug

Example request body:

```
{
  "article": {
    "title": "Did you train your dragon?"
  }
}
```

Authentication required, returns the updated Article

Optional fields: `title`, `description`, `body`

The `slug` also gets updated when the `title` is changed

Delete Article

DELETE `/api/articles/:slug`

Authentication required

Add Comments to an Article

POST `/api/articles/:slug/comments`

Example request body:

```
{
  "comment": {
    "body": "His name was my name too."
  }
}
```

Authentication required, returns the created Comment

Required fields: `body`

Get Comments from an Article

GET `/api/articles/:slug/comments`

Authentication optional, returns multiple comments

Delete Comment

DELETE `/api/articles/:slug/comments/:id`

Authentication required

Favorite Article

POST `/api/articles/:slug/favorite`

Authentication required, returns the Article

No additional parameters required

Unfavorite Article

DELETE /api/articles/:slug/favorite

Authentication required, returns the Article

No additional parameters required

Get Tags

GET /api/tags

No authentication required, returns a List of Tags