API Reference

Conduit API

API Version: 1.0.0

Conduit API

CONTACT

NAME: RealWorld
URL: https://realworld.io

INDEX

Security and Authentication

SECURITY SCHEMES

KEY	TYPE	DESCRIPTION
Token	apiKey	For accessing the protected API resources, you must have received a a valid JWT token after registering or logging in. This JWT token must then be used for all protected resources by passing it in via the 'Authorization' header. A JWT token is generated by the API by either registering via /users or logging in via /users/login. The following format must be in the 'Authorization' header : Token: xxxxxx.yyyyyyy.zzzzzz

API

1. ARTICLES

1.1 GET /articles/feed

Get recent articles from users you follow

Get most recent articles from users you follow. Use query parameters to limit. Auth is required

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
`limit`	`integer`	Limit number of articles returned (default is 20)
`offset`	`integer`	Offset/skip number of articles (default is 0)

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  articles* [{
  Array of object:
     slug*           string
     title*          string
     description*    string
     body*           string
     tagList*        [string]
     createdAt*      string
     updatedAt*      string
     favorited*      boolean
     favoritesCount* integer
     author* {
        username*  string
        bio*       string
        image*     string
        following* boolean
     }
  }]
  articlesCount*   integer
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
     body*  [string]
  }
}
```

1.2 GET /articles

Get recent articles globally

Get most recent articles globally. Use query parameters to filter results. Auth is optional

REQUEST

QUERY PARAMETERS

NAME	TYPE	DESCRIPTION
tag	string	Filter by tag
author	string	Filter by author (username)
favorited	string	Filter by favorites of a user (username)
limit	integer	Limit number of articles returned (default is 20)
offset	integer	Offset/skip number of articles (default is 0)

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  articles* [{
  Array of object:
    slug*           string
    title*          string
    description*    string
    body*           string
    tagList*        [string]
    createdAt*      string
    updatedAt*      string
    favorited*      boolean
    favoritesCount* integer
    author* {
      username*  string
      bio*       string
      image*     string
      following* boolean
    }
  }]
  articlesCount*   integer
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

1.3 POST /articles

Create an article

Create an article. Auth is required

REQUEST

REQUEST BODY - application/json

```
{
  article* {
    title*       string
    description* string
    body*        string
    tagList      [string]
  }
}
```

RESPONSE

STATUS CODE - 201: OK

RESPONSE MODEL - application/json

```
{
  article* {
    slug*           string
    title*          string
    description*    string
    body*           string
    tagList*        [string]
    createdAt*      string
    updatedAt*      string
    favorited*      boolean
    favoritesCount* integer
    author* {
      username*  string
      bio*       string
      image*     string
      following* boolean
    }
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

1.4 GET /articles/{slug}

Get an article

Get an article. Auth not required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article to get

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  article* {
    slug*           string
    title*          string
    description*    string
    body*           string
    tagList*        [string]
    createdAt*      string
    updatedAt*      string
    favorited*      boolean
    favoritesCount* integer
    author* {
      username*  string
      bio*       string
      image*     string
      following* boolean
    }
  }
}
```

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

1.5 PUT /articles/{slug}

Update an article

Update an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article to update

REQUEST BODY - application/json

```
{
  article* {
    title       string
```

```
      description string
      body        string
    }
  }
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  article* {
    slug*           string
    title*          string
    description*    string
    body*           string
    tagList*        [string]
    createdAt*      string
    updatedAt*      string
    favorited*      boolean
    favoritesCount* integer
    author* {
      username*  string
      bio*       string
      image*     string
      following* boolean
    }
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

1.6 DELETE /articles/{slug}

Delete an article

Delete an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article to delete

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

2. COMMENTS

2.1 GET /articles/{slug}/comments

Get comments for an article

Get the comments for an article. Auth is optional

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article that you want to get comments for

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   comments* [{
   Array of object:
      id*          integer
      createdAt*   string
      updatedAt*   string
      body*        string
      author* {
         username*   string
         bio*        string
         image*      string
         following*  boolean
      }
   }]
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```

2.2 POST /articles/{slug}/comments

Create a comment for an article

Create a comment for an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article that you want to create a comment for

REQUEST BODY - application/json

```
{
   comment* {
      body*  string
   }
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   comment* {
      id*          integer
      createdAt*   string
      updatedAt*   string
      body*        string
      author* {
         username*   string
         bio*        string
         image*      string
         following*  boolean
      }
   }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```

2.3 DELETE /articles/{slug}/comments/{id}

Delete a comment for an article

Delete a comment for an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article that you want to delete a comment for
*id	integer	ID of the comment you want to delete

RESPONSE

STATUS CODE - 200: OK

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

3. FAVORITES

3.1 POST /articles/{slug}/favorite

Favorite an article

Favorite an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article that you want to favorite

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  article* {
    slug*           string
    title*          string
    description*    string
    body*           string
    tagList*        [string]
    createdAt*      string
    updatedAt*      string
    favorited*      boolean
    favoritesCount* integer
    author* {
      username*  string
      bio*       string
      image*     string
      following* boolean
    }
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

3.2 DELETE /articles/{slug}/favorite

Unfavorite an article

Unfavorite an article. Auth is required

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*slug	string	Slug of the article that you want to unfavorite

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   article* {
      slug*            string
      title*           string
      description*     string
      body*            string
      tagList*         [string]
      createdAt*       string
      updatedAt*       string
      favorited*       boolean
      favoritesCount*  integer
      author* {
         username*   string
         bio*        string
         image*      string
         following*  boolean
      }
   }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```

4. PROFILE

4.1 GET /profiles/{username}

Get a profile

Get a profile of a user of the system. Auth is optional

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*username	string	Username of the profile to get

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   profile* {
      username*   string
      bio*        string
      image*      string
      following*  boolean
   }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```

4.2 POST /profiles/{username}/follow

Follow a user

Follow a user by username

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*username	string	Username of the profile you want to follow

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  profile* {
    username*  string
    bio*       string
    image*     string
    following* boolean
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

4.3 DELETE /profiles/{username}/follow

Unfollow a user

Unfollow a user by username

REQUEST

PATH PARAMETERS

NAME	TYPE	DESCRIPTION
*username	string	Username of the profile you want to unfollow

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  profile* {
    username*  string
    bio*       string
    image*     string
    following* boolean
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
```

```
        }
    }
```

5. USER AND AUTHENTICATION

5.1 POST /users/login

Existing user login

Login for existing user

REQUEST

REQUEST BODY - application/json

```
{
   user* {
      email*     string
      password*  string
   }
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   user* {
      email*     string
      token*     string
      username*  string
      bio*       string
      image*     string
   }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```

5.2 GET /users

Get current user

Gets the currently logged-in user

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  user* {
    email*    string
    token*    string
    username* string
    bio*      string
    image*    string
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

5.3 PUT /users

Update current user

Updated user information for current user

REQUEST

REQUEST BODY - application/json

```
{
  user* {
    email    string
    token    string
    username string
    bio      string
    image    string
  }
}
```

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
  user* {
    email*    string
    token*    string
    username* string
    bio*      string
    image*    string
  }
}
```

STATUS CODE - 401: Unauthorized

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

5.4 POST /users

Register a new user

REQUEST

REQUEST BODY - application/json

```
{
  user* {
    username* string
    email*    string
    password* string
  }
}
```

RESPONSE

STATUS CODE - 201: OK

RESPONSE MODEL - application/json

```
{
  user* {
    email*    string
    token*    string
    username* string
    bio*      string
    image*    string
  }
}
```

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
  errors* {
    body* [string]
  }
}
```

6. TAGS

6.1 GET /tags

Get tags

Get tags. Auth not required

REQUEST

No request parameters

RESPONSE

STATUS CODE - 200: OK

RESPONSE MODEL - application/json

```
{
   tags*  [string]
}
```

STATUS CODE - 422: Unexpected error

RESPONSE MODEL - application/json

```
{
   errors* {
      body*  [string]
   }
}
```