

User API Map

<http://localhost:3003/api/users>

create -> **POST** : /

update -> **PATCH** : [/:id](#)

delete user -> **DELETE** : [/:id\(\\d+\)](#)

[/show](#)

see all users -> **GET** : /

see user by ID -> **GET** : [/:id\(\\d+\)](#)

see user by email -> **GET** : [/:email](#)

[/testing](#)

delete in testing mode -> **DELETE** : [/:user_id\(\\d+\)](#)

[/login](#)

login -> **POST** : /

logout -> **POST** : [/:user_id](#)

Language API Map

<http://localhost:3003/api/languages>

create -> **POST** : /

update by lang id -> **PATCH**: [/:id\(\\d+\)](#)

update by ISO code -> **PATCH** : [/:isoCode\(\\\[a-zA-Z\]{2,4}\)](#)

show all -> **GET**: /

show by ISO code -> **GET**: [/:isoCode\(\\\[a-zA-Z\]{2,4}\)](#)

show by langID -> **GET**: [/:id\(\\d+\)](#)

delete by langID -> **DELETE**: [/:id\(\\d+\)](#)

delete by ISO code -> **DELETE**: [/:isoCode\(\\\[a-zA-Z\]{2,4}\)](#)

[/newCountry](#)

add new country by country ID -> **POST**: [/:id\(\\d+\)](#)

add new country by isoCode -> **POST**: [/:isoCode\(\\\[a-zA-Z\]{2,4}\)](#)

delete -> **DELETE**: /

[/alt_name](#)

search by alt name -> **GET**: [/:alt_name](#)

[/dialects](#)

get all dialects -> **GET**: /

[/continent](#)

show lang by continent -> **GET:** `/:continent`

`/region`

show lang by region -> **GET:** `/:region_name`

`/intregion`

show lang by intregion -> **GET:** `/:intregion_name`

`/country`

show lang by country -> **GET:** `/:country`

`/hdx`

Add hdx link to a country -> **POST:** `/:isoCode`

Delete hdx link from a country -> **DELETE:** `/:isoCode`

`/pt`

Add a Public Tableau link to a country -> **POST:** `/:isoCode`

Delete a Public Tableau link from a country -> **DELETE:** `/:isoCode`

`/clearglobal`

Add a CLEAR Global link to a country -> **POST:** `/:isoCode`

Delete a CLEAR Global link from a country -> **DELETE:** `/:isoCode`

`/requests`

create a new lang request -> **POST:** `/`

update a request by id -> **PATCH:** `/:id`

show requests by id -> **GET:** `/:id(\\d+)`

show all requests -> **GET:** `/`

`/complete`

show all completed requests -> **GET:** `/`

show completed requests by ID -> **GET:** `/:id(\\d+)`

`/open`

show all open requests -> **GET:** `/`

show open requests by ID -> **GET:** `/:id(\\d+)`

`/pending`

show pending requests -> **GET:** `/`

show pending requests by ID -> **GET:** `/:id(\\d+)`

`/refs`

Add references to language -> **POST:** `/:id`

Update reference to language -> **PATCH:** `/:id`

Delete reference from language -> **DELETE:** `/:id`

`/sourceComment`

Add source comment to language -> **POST:** [/:id](#)

Delete source comment from language -> **DELETE:** [/](#)

[/alternativeNames](#)

Add alternative name to language -> **POST:** [/:id](#)

Delete alternative name from language -> **DELETE:** [/](#)

[/links](#)

Add links to languages -> **POST:** [/:id](#)

Update links to language -> **PATCH:** [/:id](#)

Delete links from language -> **DELETE:** [/:id](#)

App API Map

<http://localhost:3003/api/app>

[/countries](#)

Show all countries' info-> **GET:** [/](#)

Show names of countries speaking a certain language -> **GET:** [/:lang](#)
[/info](#)

Show all information about country -> **GET:** [/:country_name](#)

[/lang](#)

Show all languages and essential details about them -> **GET:** [/](#)

Show details of one language in particular -> **GET:** [/:lang](#)

[/langNames](#)

Show names of all languages -> **GET:** [/](#)

[/langDetails](#)

Show all relevant details for all languages -> **GET:** [/](#)

[/requests](#)

Show requests by language name -> **GET:** [/:lang](#)

[/period](#)

Show all requests between certain dates -> **GET:** [/:lang](#)

[/recent](#)

Show most recent request -> **GET:** [/](#)

[/password](#)

Sends a token and timestamp used to reset password -> **POST:** [/](#)

Extracts the token used to reset password -> **GET:** [/](#)

Every query can handle internal server error.

For example this is the response when a package json file hasn't been compiled correctly:

500 Internal Server Error	
log	{"code":"ER_BAD_NULL_ERROR","errno":1048,"level":"error","sql":"...", "sqlState":"23000"}
res	Database connection error

All queries that **require user tokens** to use will return error 401 if no token is provided and error 400 if an invalid token is provided.

400 Invalid authorization token	
log	Access denied. Invalid token
res	Invalid email or password

401 No authorization token	
log	Access denied. Unauthorized user.
res	Access denied. Unauthorized user.

Create User (createUser)

GET(http://localhost:3003/api/users)

Request package

```
{
  "first_name": "Fedor",
  "last_name": "Test",
  "email": "testing@mail.com",
  "password": "1234",
  "hash": "bruh",
  "mobile": "123456",
  "user_type_id": 1,
  "position": "",
  "company": "",
  "user_image": "topkek.jpg",
  "status": 1,
  "address": "10 Downing St.",
  "postcode": "SW1A 2AA"
}
```

Responses

200 Success	
log	New user created
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 2653, "info": "", "serverStatus": 2, "warningStatus": 0 }
400 User with this email is already registered	
log	User with this email is already registered: '+ email
res	User with this email is already registered: '+ email

Notes

None

Update User (updateUser)

PATCH(<http://localhost:3003/api/users/:id>)

id = user_id from users

Request package

```
{
  "first_name": "Katharine",
  "last_name": "Test",
  "email": "testing@mail.com",
  "password": "1234",
  "hash": "bruh",
  "mobile": "123456",
  "user_type_id": 1,
  "position": "",
  "company": "",
  "user_image": "topkek.jpg",
  "status": 1
}
```

Responses

200 Success

log	'User updated. User id: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 3", "serverStatus": 2, "warningStatus": 3, "changedRows": 1 }

200 No changes implemented

log	'No content has been changed. User id: '+id
res	No changes implemented

404 Could not find user

log	'Could not find user. User id: '+id
res	Could not find user

Notes

User token checked to use

Delete User (deleteUsers)

```
DELETE( http://localhost:3003/api/users/:id\(\\d+\) )  
      id = user_id from users
```

Request package

None

Responses

200 Success	
log	'User deleted. User id: '+ id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 3", "serverStatus": 2, "warningStatus": 3, "changedRows": 1 }
200 No changes implemented	
log	'No content has been changed. User id: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }
404 Could not find user	
log	Could not find user
res	'Could not find user. User id: '+id

Notes

User token checked to use

Show Users (showAllUsers)

GET(<http://localhost:3003/api/users/show/>)

Request package

None

Responses

200 Success	
log	results.length+' users found'
res	<pre>[{ "user_id": 2, "first_name": "Anelia", "last_name": "Gaydardzhieva", "email": "aneliakg@protonmail.com", "password": "\$2y\$12\$3kfJelbtd0qlXFTFxdoHn.vFogKF4aB9SA4SppSie/VayJuEJt.SK", "hash": "VWIsyLjxOVuSLzXkpJtY", "mobile": "aneliakg@protonmail.com", "user_type_id": 1, "position": "", "company": "", "joined_date": 1658689183, "user_image": "anelia.jpg", "status": 1, "address": "", "postcode": "" }, ...]</pre>

Notes

None

Show User (showUserByID)

GET([http://localhost:3003/api/users/show/:id\(\\d+\)](http://localhost:3003/api/users/show/:id(\\d+)))

id = user_id from users

Request package

None

Responses

200 Success	
log	'User found. User id: '+id
res	{ "user_id": 2, "first_name": "Anelia", "last_name": "Gaydardzhieva", "email": "aneliakg@protonmail.com", "password": "\$2y\$12\$3kfJelbtd0q1XFTFxdohn.vFogKF4aB9SA4SppSie/VayJuEJt.SK", "hash": "VWIsyLjxOVuSLzXkpJtY", "mobile": "aneliakg@protonmail.com", "user_type_id": 1, "position": "", "company": "", "joined_date": 1658689183, "user_image": "anelia.jpg", "status": 1, "address": "", "postcode": "" }
404 Could not find user	
log	'Could not find user. User id: '+id
res	Could not find user

Notes

None

Show User by Email (showUserByEmail)

GET(<http://localhost:3003/api/users/show/:email>)

Request package

None

Responses

200 Success	
log	'User found. User email: '+email
res	{ "user_id": 2, "first_name": "Anelia", "last_name": "Gaydardzhieva", "email": "aneliakg@protonmail.com", "password": "\$2y\$12\$3kfJelbtd0q1XFTFxdohn.vFogKF4aB9SA4SppSie/VayJuEJt.SK", "hash": "VWIsyLjxOVuSLzXkpJtY", "mobile": "aneliakg@protonmail.com", "user_type_id": 1, "position": "", "company": "", "joined_date": 1658689183, "user_image": "anelia.jpg", "status": 1, "address": "", "postcode": "" }
404 Could not find user	
log	'Could not find user. User id: '+email
res	Could not find user

Notes

None

Delete Test History (deleteTestHistory)

```
DELETE( http://localhost:3003/api/users/testing/:user\_id\(\\d+\) )  
      user_id = user_id from history
```

Request package

None

Responses

200	Success
log	N/A
res	'deleted'

Notes

User token checked to use

Login (login)

POST(<http://localhost:3003/api/users/login>)

Request package

```
{
  "email": "testing@mail.com",
  "password": "1234"
}
```

Responses

200	Success
log	'Login successful: user ' + user_id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 520, "info": "", "serverStatus": 2, "warningStatus": 0, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. eyJyZXNlbHQiOnsidXNlcl9pZCI6MjY1NCwiZmlyczRfbmFtZSI6IkZlZG9yIiwibGFzdF9uYW1lIjoiiVGVzdCIsImVtYWlsIjoidGVzdGluZ0BtYWlsLmNvbSIsImhhcg2giOiJicnVoIiwibW9iaWxlIjoiiMTIzNDU2IiwidXNlcl90eXB1X2lkIjoixLCJwb3NpdGlubiI6IiIsImNvbXBhbHkiOiIiLCJqb2luZWRFfZGF0ZSI6MTY4MTkzMjgOMywidXNlcl9pbWFhZSI6InRvcGtlays5qcGciLCJzdGF0dXMiojEsImFkZHJlc3MiOiIxMCBEb3duaW5nIFN0LiIsInBvc3Rjb2RlIjoiiUlcnQSAyQUEifSwiaWF0IjoxNjgxOTMyODQ2LCJleHAiOjE2ODE5MzY0NDZ9.b4b_HzacexmWGPFftO9mxqFISv9FPbTL_vwycS2Cjuj4" }

404	Could not find user
log	Invalid password - results returned
res	Invalid email or password

404	Could not find user - no results returned by query
log	Invalid email / No details provided - no results returned
res	Invalid email or password

Notes

None

Logout (logout)

POST(http://localhost:3003/api/users/login/:user_id)
user_id = user_id from history

Request package

None

Responses

200	Success
log	'Logout successful: user ' + user_id
res	logout successful!

Notes

None

Create Language (createLang)

POST(<http://localhost:3003/api/languages/>)

Request package

```
{
  "source_id":1,
  "lang_name":"The testenese",
  "iso_code":"TEST",
  "no_of_trans":1,
  "lang_status":"verified",
  "glottocode":"test1234",
  "total_speakers_nr":"test",
  "first_lang_speakers_nr":"test",
  "second_lang_speakers_nr":"test",
  "level":"language",
  "aes_status":"test",
  "family_name":"test"
}
```

Responses

200 Success	
log	New language created
res	<pre>{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 1 }</pre>
400 User with this email is already registered	
log	<pre>{"code":"ER_DUP_ENTRY","errno":1062,"level":"error","sql":"...","sqlState":"23000"}</pre>
res	Language with isoCode <code>\${iso_code}</code> already exists

Notes

User token checked to use

Update Language (updateLangByID)

PATCH([http://localhost:3003/api/languages/:id\(\\d+\)](http://localhost:3003/api/languages/:id(\\d+)))

id = lang_id from languages

Request package

```
{
  "source_id":1,
  "lang_name":"The Updated Test",
  "iso_code":"tst",
  "no_of_trans":1,
  "lang_status":1,
  "glottocode":"test",
  "total_speakers_nr":"test",
  "first_lang_speakers_nr":"test",
  "second_lang_speakers_nr":"test",
  "level":"language",
  "aes_status":"test",
  "family_name":"test"
}
```

Responses

200 Success	
log	'Language updated. Language ID: '+ id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 1", "serverStatus": 2, "warningStatus": 1, "changedRows": 1 }

200 No changes implemented	
log	'No content has been changed. Language ID code: '+id
res	No changes implemented

404 Could not find language	
log	"Could not find language. Language ID code: "+id
res	Could not find language

Notes

User token checked to use

Update Language by ISO Code (updateLangByISO)

PATCH([http://localhost:3003/api/languages/:isoCode\(\\[a-zA-A\]{2,4}\)](http://localhost:3003/api/languages/:isoCode(\[a-zA-A]{2,4})))

isoCode = iso_code from languages

Request package

```
{
  "source_id":1,
  "lang_name":"The Updated Test",
  "iso_code":"TEST",
  "no_of_trans":1,
  "lang_status":1,
  "glottocode":"test",
  "total_speakers_nr":"test",
  "first_lang_speakers_nr":"test",
  "second_lang_speakers_nr":"test",
  "level":"language",
  "aes_status":"test",
  "family_name":"test"
}
```

Responses

200 Success	
log	'Language updated. ISO code: '+isoCode
res	{ "fieldCount": 0, "affectedRows": 2, "insertId": 0, "info": "Rows matched: 2 Changed: 2 Warnings: 2", "serverStatus": 34, "warningStatus": 2, "changedRows": 2 }

200 No changes implemented	
log	'No content has been changed. Language ISO code: '+isoCode
res	No changes implemented

404 Could not find language	
log	"Could not find language. Language ISO code: "+isoCode
res	Could not find language

Notes

User token checked to use

Show Languages (showAll)

GET(<http://localhost:3003/api/languages/>)

Request package

None

Responses

200	Success
log	results.length+' languages found'
res	<pre>[{ "language": { "lang_id": 3, "source_id": 3, "lang_name": "Balochi", "iso_code": "bcc", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "not endangered", "family_name": "Indo-European" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "https://phoible.org/languages/sout2642", "description": "PHOIBLE" }, ...], "source_comments": ["David M. Eberhard and Gary F. Simons and Charles D. Fennig 2022"], "refs": [] }, ...]</pre>

Notes

None

Show language by ISO Code (showLangByISO)

GET([http://localhost:3003/api/languages/:isoCode\(\\[a-zA-Z\]{2,4}\)](http://localhost:3003/api/languages/:isoCode(\[a-zA-Z]{2,4})))

isoCode = iso_code from languages

Responses

200 Success	
log	'Language found. ISO code: '+ isoCode
res	<pre>{ "language": { "lang_id": 3, "source_id": 3, "lang_name": "Balochi", "iso_code": "bcc", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "not endangered", "family_name": "Indo-European" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "https://phoible.org/languages/sout2642", "description": "PHOIBLE" }, ...], "source_comments": ["David M. Eberhard and Gary F. Simons and Charles D. Fennig 2022"], "refs": [] }</pre>

404 Could not find language	
log	'Could not find language. Language ISO code: '+isoCode
res	Could not find language

Show language (showLangByID)

GET([http://localhost:3003/api/languages/:id\(\\d+\)](http://localhost:3003/api/languages/:id(\\d+)))

id = lang_id from languages

Responses

200 Success	
log	'Language found. Language ID: '+ id
res	<pre>{ "language": { "lang_id": 3, "source_id": 3, "lang_name": "Balochi", "iso_code": "bcc", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "not endangered", "family_name": "Indo-European" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "https://phoible.org/languages/sout2642", "description": "PHOIBLE" }, ...], "source_comments": ["David M. Eberhard and Gary F. Simons and Charles D. Fennig 2022"], "refs": [] }</pre>
404 Could not find language	
log	'Could not find language. Language ID: '+ id
res	Could not find language

Delete Language (deleteLangByID)

```
DELETE( http://localhost:3003/api/languages/:id\(\\d+\) )  
      id = lang_id from languages
```

Request package

None

Responses

200 Success	
log	'Language deleted. ID code: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 Could not find language	
log	Could not find language. ID code: '+ id
res	Could not find language

Notes

User token checked to use

Delete Language by ISO Code (deleteLangByISO)

```
DELETE( http://localhost:3003/api/languages/:isoCode\(\\[a-zA-Z\]{2,4}\) )  
      isoCode = iso_code from languages
```

Request package

None

Responses

200 Success	
log	'Language deleted. ISO code: '+isoCode
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 Could not find language	
log	Could not find language. ISO code: '+isoCode
res	Could not find language

Notes

User token checked to use

Add new country to language (addCountryToLanguageByID)

POST([http://localhost:3003/api/languages/newCountry/:id\(\\d+\)](http://localhost:3003/api/languages/newCountry/:id(\\d+)))

id = lang_id from languages

Request package

```
{
  "country_iso_code": "AF",
  "official": 1,
  "national": 0,
  "internet_users_percent": 10
}
```

Responses

200 Success	
log	'Language location updated. Country ID: '+country_iso_code+'. Language ID: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Language is already paired with that country	
log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	Language with ID \${id} and country ID \${country_iso_code} already exists

Notes

User token checked to use

Add new country by ISO Code (addCountryToLanguageByISO)

POST([http://localhost:3003/api/languages/newCountry/:isoCode\(\\[a-zA-Z\]{2,4}\)](http://localhost:3003/api/languages/newCountry/:isoCode(\[a-zA-Z]{2,4})))
isoCode = iso_code from languages

Request package

```
{  
  "country_iso_code": "AF",  
  "official": 1,  
  "national": 0,  
  "internet_users_percent": 10  
}
```

Responses

200 Success	
log	Language location updated. Country ID: '+country_iso_code+'. ISO code: '+isoCode'
res	<pre>{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 34, "warningStatus": 0 }</pre>

400 Language is already paired with that country	
log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	Language with ISO Code \${isoCode} and country ID \${country_iso_code} already exists

Notes

User token checked to use

Remove country from language (deleteLangsCountry)

DELETE(<http://localhost:3003/api/languages/newCountry>)

Request package

```
{  
  "lang_id":0,  
  "country_iso_code":"AF"  
}
```

Responses

200 Success	
log	'Record deleted. Language ID: '+lang_id+'. Country ID: '+country_iso_code
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 Could not find record	
log	'Could not find record. Language ID: '+lang_id+'. Country ID: '+country_iso_code
res	Could not find record

Notes

User token checked to use

Search by Alternative Name (showByAltName)

GET(http://localhost:3003/api/languages/alt_name/:alt_name)

alt_name = alternative_name from alternative_names

Responses

200 Success	
log	'Language found. Alternative name: '+alt_name
res	[{ "language": { "lang_id": 3, "source_id": 3, "lang_name": "Balochi", "iso_code": "bcc", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "not endangered", "family_name": "Indo-European" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "http://odin.linguistlist.org/igt_urls.php?lang=bcc", "description": "[bcc] at Odin" }, ...], "source_comments": ["David M. Eberhard and Gary F. Simons and Charles D. Fennig 2022"], "refs": [] }]
404 Could not find record	
log	'Could not find language. Language alternative name: '+ alt_name
res	Could not find record

Notes

It returns a list of json queries in case some languages share the same alternative names.

Show all Dialects (showAllDialects)

GET(<http://localhost:3003/api/languages/dialects>)

Request package

None

Responses

200 Success	
log	'Record deleted. Language ID: '+lang_id+'. Country ID: '+country_iso_code
res	[{ "language": { "lang_id": 1479, "source_id": 2, "lang_name": "3Ad-Tekles", "iso_code": "", "no_of_trans": null, "lang_status": "in progress", "glottocode": "3adt1234", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "dialect", "aes_status": "", "family_name": "Afro-Asiatic" }, "alternative_names": [null], "links": [], "source_comments": [], "refs": [] }, ...]

Notes

None

Show languages in continent (showLanguagesByContinent)

GET(<http://localhost:3003/api/languages/continent/:continent>)

continent = continent from regions_continents

Request package

None

Responses

200	Success
log	results.length + ' languages found for continent: ' + continent
res	<pre>[{ "language": { "lang_id": 1479, "source_id": 2, "lang_name": "3Ad-Tekles", "iso_code": "", "no_of_trans": null, "lang_status": "in progress", "glottocode": "3adt1234", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "dialect", "aes_status": "", "family_name": "Afro-Asiatic" }, "alternative_names": [null], "links": [], "source_comments": [], "refs": [] }, ...]</pre>
404	Could not find record
log	'Could not find languages for continent: ' + continent
res	Could not find languages

Notes

There is no distinction between South America and North America in the database

Show languages in region (showLanguagesByRegion)

GET(http://localhost:3003/api/languages/region/:region_name)

region_name = regions from regions_continents

Request package

None

Responses

200	Success
log	results.length + ' languages found for region: ' + region_name
res	<pre>[{ "language": { "lang_id": 1479, "source_id": 2, "lang_name": "3Ad-Tekles", "iso_code": "", "no_of_trans": null, "lang_status": "in progress", "glottocode": "3adt1234", "total_speakers_nr": "", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "dialect", "aes_status": "", "family_name": "Afro-Asiatic" }, "alternative_names": [null], "links": [], "source_comments": [], "refs": [] }, ...]</pre>
404	Could not find record
log	'Could not find languages for region: ' + region_name
res	Could not find languages

Notes

None

Show languages in int. region (showLanguagesByIntregion)

GET(http://localhost:3003/api/languages/intregion/:intregion_name)

intregion_name = intermediate_regions from countries_regions_int

Request package

None

Responses

200 Success	
log	results.length + ' languages found for intermediate region: ' + intregion_name
res	<pre>[{ "language": { "lang_id": 2, "source_id": 1, "lang_name": "Arabic", "iso_code": "ara", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "200-500 million", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "", "family_name": "" }, "alternative_names": [null], "links": [], "source_comments": [], "refs": [] }, ...]</pre>

404 Could not find record	
log	'Could not find languages for intermediate region: ' + intregion_name
res	Could not find languages

Notes

None

Show languages in country (showLanguagesByCountry)

GET(<http://localhost:3003/api/languages/country/:country>)

country = english_name from countries

Request package

None

Responses

200 Success	
log	results.length + ' languages found for country: ' + country
res	<pre>[{ "language": { "lang_id": 2, "source_id": 1, "lang_name": "Arabic", "iso_code": "ara", "no_of_trans": null, "lang_status": "in progress", "glottocode": "", "total_speakers_nr": "200-500 million", "first_lang_speakers_nr": "", "second_lang_speakers_nr": "", "level": "language", "aes_status": "", "family_name": "" }, "alternative_names": [null], "links": [], "source_comments": [], "refs": [] }, ...]</pre>

404 Could not find record	
log	'Could not find languages for country: ' + country
res	Could not find languages

Notes

None

Add HDX link to a country (addHDXLinks)

POST(<http://localhost:3003/api/languages/hdx/:isoCode>)

isoCode = country_iso_code from countries

Request package

```
{
  "link": "google.com",
  "description": "excellent research tool"
}
```

Responses

200 Success

log	'New links added, for country with iso_code: '+isoCode
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Repeated link for country

log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Links for country with iso_code: \${isoCode} already exist`

404 Could not find record

log	{"code": "ER_NO_REFERENCED_ROW_2", "errno": 1452, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Country with id \${isoCode} not found`

Notes

User token checked to use

Remove HDX link from a country (deleteHDXLink)

```
DELETE( http://localhost:3003/api/languages/hdx/:isoCode )  
      isoCode = country_iso_code from countries
```

Request package

```
{  
  "link": "google.com"  
}
```

Responses

200 Success	
log	`Link deleted. Link: \${link}. Country ISO Code: \${isoCode}`
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 Could not find record	
log	`Could not find link. Link: \${link}. Country ISO Code: \${isoCode}`
res	Could not find link

Notes

User token checked to use

Add Public Tableau link to a country (addPublicTableauLinks)

POST(<http://localhost:3003/api/languages/pt/:isoCode>)

isoCode = country_iso_code from countries

Request package

```
{
  "link": "google.com",
  "description": "excellent research tool"
}
```

Responses

200 Success	
log	'New links added, for country with iso_code: '+isoCode
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Repeated link for country	
log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Links for country with iso_code: \${isoCode} already exist`

404 Could not find record	
log	{"code": "ER_NO_REFERENCED_ROW_2", "errno": 1452, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Country with id \${isoCode} not found`

Notes

User token checked to use

Remove Public Tableau link from a country (deletePublicTableauLink)

```
DELETE( http://localhost:3003/api/languages/pt/:isoCode )  
      isoCode = country_iso_code from countries
```

Request package

```
{  
  "link": "google.com"  
}
```

Responses

200 Success

log	`Link deleted. Link: \${link}. Country ISO Code: \${isoCode}`
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 Could not find record

log	`Could not find link. Link: \${link}. Country ISO Code: \${isoCode}`
res	Could not find link

Notes

User token checked to use

Add CLEAR Global link to a country (addClearGlobalLinks)

POST(<http://localhost:3003/api/languages/clearglobal/:isoCode>)

isoCode = country_iso_code from countries

Request package

```
{
  "link": "google.com",
  "description": "excellent research tool"
}
```

Responses

200 Success	
log	'New links added, for country with iso_code: '+isoCode
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Repeated link for country	
log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Links for country with iso_code: \${isoCode} already exist`

404 Could not find record	
log	{"code": "ER_NO_REFERENCED_ROW_2", "errno": 1452, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Country with id \${isoCode} not found`

Notes

User token checked to use

Remove CLEAR Global link from a country (deleteClearGlobalLink)

```
DELETE( http://localhost:3003/api/languages/clearglobal/:isoCode )  
      isoCode = country_iso_code from countries
```

Request package

```
{  
  "link": "google.com"  
}
```

Responses

200 Success	
log	`Link deleted. Link: \${link}. Country ISO Code: \${isoCode}`
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }
404 Could not find record	
log	`Could not find link. Link: \${link}. Country ISO Code: \${isoCode}`
res	Could not find link

Notes

User token checked to use

Create new suggestion for language (createLangRequests)

POST(<http://localhost:3003/api/languages/requests/>)

Request package

```
{
  "assigned_user_id":0,
  "lang_id":0,
  "lr_type":"add",
  "lr_title":"Test",
  "lr_reason":"I need to test",
  "lr_lang_name":"Testenese",
  "lr_alternative_name":"",
  "lr_iso_code":"",
  "lr_glottocode":"",
  "lr_added_countries":"",
  "lr_removed_countries":"",
  "lr_lang_status":"verified",
  "lr_status":"complete"
}
```

Responses

200 Success	
log	`New language request created. ID: \${insertId}. Language: \${lang_id}. Assigned user: \${assigned_user_id}`
res	<pre>{ "fieldCount": 0, "affectedRows": 1, "insertId": 916, "info": "", "serverStatus": 2, "warningStatus": 2 }</pre>

Notes

User token checked to use

Update suggestion for language (updateLangRequests)

PATCH(<http://localhost:3003/api/languages/requests/:id>)

id = lang_request_id from language_requests

Request package

```
v{
  "assigned_user_id":0,
  "lang_id":0,
  "lr_type":"add",
  "lr_title":"Test",
  "lr_reason":"I need to test",
  "lr_lang_name":"Testenese",
  "lr_alternative_name":"",
  "lr_iso_code":"",
  "lr_glottocode":"",
  "lr_added_countries":"",
  "lr_removed_countries":"",
  "lr_lang_status":"verified",
  "lr_status":"complete"
}
```

Responses

200 Success	
log	'Language request updated. Language ID: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 6", "serverStatus": 2, "warningStatus": 6, "changedRows": 1 }
200 No content changed	
log	'No content has been changed. Request ID code: '+id
res	No changes implemented
404 Could not find request	
log	"Could not find request. Request ID code: "+id
res	Could not find request

Notes

User token checked to use

Show suggestion by request ID (showLangRequestsByID)

GET([http://localhost:3003/api/languages/requests/:id\(\\d+\)](http://localhost:3003/api/languages/requests/:id(\\d+)))

id = lang_request_id from language_requests

Request package

None

Responses

200 Success	
log	'Request found. Language ID: '+id
res	[{ "lang_request_id": 924, "created_user_id": 0, "assigned_user_id": 0, "lr_end_date": 0, "lr_start_date": 0, "lang_id": 0, "lr_type": "add", "lr_title": "Test", "lr_reason": "I need to test", "lr_lang_name": "Testenese", "lr_alternative_name": "Teston", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "complete" }]
404 Could not find request	
log	'Could not find request. Language ID: '+id
res	Could not find request

Notes

User token checked to use

Show all suggestions (showAllRequests)

GET(<http://localhost:3003/api/languages/requests/>)

Request package

None

Responses

200 Success	
log	results.length+' requests found'
res	<pre>[{ "lang_request_id": 6, "created_user_id": 2, "assigned_user_id": 0, "lr_end_date": 0, "lr_start_date": 1664281504, "lang_id": 511, "lr_type": "edit", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "pending" }, ...]</pre>

Notes

User token checked to use

Show all complete suggestions (showAllCompleteRequests)

GET(<http://localhost:3003/api/languages/requests/complete/>)

Request package

None

Responses

200 Success	
log	results.length+' requests found'
res	<pre>[{ "lang_request_id": 8, "created_user_id": 15, "assigned_user_id": 19, "lr_end_date": 0, "lr_start_date": 1664866657, "lang_id": 477, "lr_type": "add", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "complete" }, ...]</pre>

Notes

None

Show complete requests by language (showCompleteRequestByLang)

GET([http://localhost:3003/api/languages/requests/complete/:id\(\\d+\)](http://localhost:3003/api/languages/requests/complete/:id(\\d+)))

id = lang_id from languages

Request package

None

Responses

200	Success
log	results.length + ' complete requests found for ID: ' + id
res	[{ "lang_request_id": 8, "created_user_id": 15, "assigned_user_id": 19, "lr_end_date": 0, "lr_start_date": 1664866657, "lang_id": 477, "lr_type": "add", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "complete" }, ...]

404	Could not return complete request history.
log	'Could not find complete request history for language ID: ' + id
res	Could not return complete request history.

Notes

None

Show all open requests (showAllOpenRequests)

GET(<http://localhost:3003/api/languages/requests/open/>)

Request package

None

Responses

200 Success	
log	results.length + ' open requests.'
res	<pre>[{ "lang_request_id": 5, "created_user_id": 2, "assigned_user_id": 19, "lr_end_date": 0, "lr_start_date": 1661976021, "lang_id": 510, "lr_type": "edit", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "in progress" }, ...]</pre>

Notes

User token checked to use

Show open requests by language (showOpenRequestByLang)

GET([http://localhost:3003/api/languages/requests/open/:id\(\\d+\)](http://localhost:3003/api/languages/requests/open/:id(\\d+)))

id = lang_id from languages

Request package

None

Responses

200 Success	
log	results.length + ' open requests found for ID: ' + id
res	<pre>[{ "lang_request_id": 5, "created_user_id": 2, "assigned_user_id": 19, "lr_end_date": 0, "lr_start_date": 1661976021, "lang_id": 510, "lr_type": "edit", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "in progress" }, ...]</pre>

404 Could not return open request history.	
log	'Could not find open request history for language ID: ' + id
res	Could not return open request history.

Notes

User token checked to use

Show all pending requests (showAllPendingRequests)

GET(<http://localhost:3003/api/languages/requests/pending/>)

Request package

None

Responses

200	Success
log	results.length + ' pending requests.'
res	<pre>[{ "lang_request_id": 6, "created_user_id": 2, "assigned_user_id": 0, "lr_end_date": 0, "lr_start_date": 1664281504, "lang_id": 511, "lr_type": "edit", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "pending" }, ...]</pre>

Notes

User token checked to use

Show pending requests by language (showPendingRequestByLang)

GET([http://localhost:3003/api/languages/requests/pending/:id\(\\d+\)](http://localhost:3003/api/languages/requests/pending/:id(\\d+)))

id = lang_id from languages

Request package

None

Responses

200	Success
log	results.length + ' pending requests found for ID: ' + id
res	[{ "lang_request_id": 5, "created_user_id": 2, "assigned_user_id": 19, "lr_end_date": 0, "lr_start_date": 1661976021, "lang_id": 510, "lr_type": "edit", "lr_title": "", "lr_reason": "", "lr_lang_name": "", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "in progress" }, ...]
404	Could not return pending request history.
log	'Could not find pending request history for language ID: ' + id
res	Could not return pending request history.

Notes

User token checked to use

Add reference to language (addRefs)

POST(<http://localhost:3003/api/languages/refs/:id>)

id = lang_id from languages

Request package

```
{
  "glottolog_ref_id":123456789,
  "lgcode":"test",
  "bib":"test"
}
```

Responses

200 Success

log 'New references added, for language with id: '+id

res [

```
{
  "fieldCount": 0,
  "affectedRows": 0,
  "insertId": 0,
  "info": "",
  "serverStatus": 16395,
  "warningStatus": 0,
  "stateChanges": {
    "systemVariables": {},
    "schema": null,
    "trackStateChange": null
  }
},
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 16678,
  "info": "",
  "serverStatus": 11,
  "warningStatus": 0
},
{
  "fieldCount": 0,
  "affectedRows": 1,
  "insertId": 0,
  "info": "",
  "serverStatus": 11,
  "warningStatus": 0
},
{
  "fieldCount": 0,
  "affectedRows": 0,
  "insertId": 0,
  "info": ""
}
```

```

    "serverStatus": 16386,
    "warningStatus": 0,
    "stateChanges": {
      "systemVariables": {},
      "schema": null,
      "trackStateChange": null
    }
  }
]

```

404 Language not found

log	{ "code": "ER_NO_REFERENCED_ROW_2", "errno": 1452, "level": "error", "sql": "...", "sqlState": "23000" }
res	`Language with id \${id} not found`

Notes

User token checked to use

Update reference to language (updateRefs)

PATCH(<http://localhost:3003/api/languages/refs/:id>)
 id = ref_id from refs

Request package

```

{
  "glottolog_ref_id": 12345678,
  "lgcode": "test",
  "bib": "test"
}

```

Responses

200 Success

log	'Reference updated. Reference ID: '+id
res	<pre> { "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 0", "serverStatus": 3, "warningStatus": 0, "changedRows": 1 } </pre>

200	No changes implemented
log	'No content has been changed. Reference ID code: '+id
res	No changes implemented

404	No reference found
log	"Could not find reference. Reference ID code: "+id
res	Could not find reference

Notes

User token checked to use

Delete reference to language (deleteRefs)

DELETE(<http://localhost:3003/api/languages/refs/:id>)
id = ref_id from refs

Responses

200	Success
log	'Reference deleted. ID: '+id
res	{ "fieldCount": 0, "affectedRows": 2, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404	No reference found
log	"Could not find reference. Reference ID code: "+id
res	Could not find reference

Notes

User token checked to use

Add source comment to language (addSourceComment)

POST(<http://localhost:3003/api/languages/sourceComment/:id>)

id = lang_id from languages

Request package

```
{  
  "comment": "Fedor Sulitskiy, Waris Hezb, Daniel Larbi, Katharine Sparks"  
}
```

Responses

200 Success

log	'New source comment added, for language with id: '+id
-----	---

res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }
-----	---

400 Source comment for language already exists

log	{"code":"ER_DUP_ENTRY","errno":1062,"level":"error","sql":"...","sqlState":"23000"}
-----	---

res	`Source comment \${comment} for language with id: \${id} already exists`
-----	--

404 Could not find language

log	{"code":"ER_NO_REFERENCED_ROW_2","errno":1452,"level":"error","sql":"...","sqlState":"23000"}
-----	---

res	`Language with id \${id} not found`
-----	-------------------------------------

Notes

User token checked to use

Delete source comment from language (deleteSourceComment)

DELETE(<http://localhost:3003/api/languages/sourceComment/>)

Request package

```
{
  "lang_id":0,
  "comment":"Fedor Sulitskiy, Waris Hezb, Daniel Larbi, Katharine Sparks"
}
```

Responses

200 Success	
log	`Source comment deleted. Language ID: \${lang_id}. Source Comment: \${comment}`
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }
404 No source comment found	
log	`Could not find source comment. Language ID: \${lang_id}. Source Comment: \${comment}`
res	Could not find source comment

Notes

User token checked to use

Add alternative name to language (addAlternativeName)

POST(<http://localhost:3003/api/languages/alternativeNames/:id>)

id = lang_id from languages

Request package

```
{
  "alternative_name": "altenative test name",
  "source": "I made it up :))"
}
```

Responses

200 Success

log	'New alternative names added, for language with id: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Alternative name for language already exists

log	{"code":"ER_DUP_ENTRY","errno":1062,"level":"error","sql":"...","sqlState":"23000"}
res	`Alternative name \${alternative_name} for language with id: \${params.id} already exist`

404 Could not find language

log	{"code":"ER_NO_REFERENCED_ROW_2","errno":1452,"level":"error","sql":"...","sqlState":"23000"}
res	`Language with id \${id} not found`

Notes

User token checked to use

Delete alternative language from language (deleteAlternativeLanguage)

DELETE(<http://localhost:3003/api/languages/alternativeNames/>)

Request package

```
{
  "lang_id":0,
  "alternative_name":"altenative test name",
  "source":"I made it up :))"
}
```

Responses

200 Success	
log	`Alternative name deleted. Language name: \${lang_id}. Alternative name: \${alternative_name}. Source: \${source}`
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }

404 No alternative name found	
log	`Could not find alternative name. Language name: \${lang_id}. Alternative name: \${alternative_name}. Source: \${source}`
res	Could not find alternative name

Notes

User token checked to use

Add link to language (addLinks)

POST(<http://localhost:3003/api/languages/links/:id>)

id = lang_id from languages

Request package

```
{
  "link": "https://www.youtube.com/watch?v=dQw4w9WgXcQ",
  "description": "it is a rick roll"
}
```

Responses

200 Success	
log	'New links added, for language with id: '+req.params.id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 58, "info": "", "serverStatus": 2, "warningStatus": 0 }

400 Link for language exists	
log	{"code": "ER_DUP_ENTRY", "errno": 1062, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Language with id \${id} not found`

404 Language not found	
log	{"code": "ER_NO_REFERENCED_ROW_2", "errno": 1452, "level": "error", "sql": "...", "sqlState": "23000"}
res	`Language with id \${id} not found`

Notes

User token checked to use

Update link (updateLinks)

PATCH(<http://localhost:3003/api/languages/links/:id>)

id = link_id from links

Request package

```
{
  "lang_id":12,
  "link":"open.ai",
  "description":"very good"
}
```

Responses

200 Success	
log	'Link updated. Link ID: '+id
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "Rows matched: 1 Changed: 1 Warnings: 0", "serverStatus": 3, "warningStatus": 0, "changedRows": 1 }

200 No changes implemented	
log	'No content has been changed. Link ID: '+req.params.id
res	No changes implemented

404 No reference found	
log	'Could not find link. Link ID: '+req.params.id
res	Could not find link

Notes

User token checked to use

Delete link from language (deleteLink)

DELETE(<http://localhost:3003/api/languages/links/:id>)

id = link_id from links

Responses

200 Success	
log	'Link deleted. Link ID: '+id
res	<pre>{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 2, "warningStatus": 0 }</pre>

404 No link found	
log	'Could not find link. Link ID: '+id
res	Could not find link

Notes

User token checked to use

Show countries information (showAllCountries)

GET(<http://localhost:3003/api/app/countries/>)

Request package

None

Responses

200 Success	
log	results.length+' countries found'
res	<pre>[{ "country_iso_code": "AU", "english_name": "Australia", "regions": "Australia and New Zealand", "intermediate_regions": "N/A", "continents": "Oceania", "hdx_link": null, "hdx_description": null, "pt_link": null, "pt_description": null, "cg_link": null, "cg_description": null }, ...]</pre>

Notes

None

Show names countries by language (showCountriesByLang)

GET(<http://localhost:3003/api/app/countries/:lang>)

lang = lang_name from languages

Request package

None

Responses

200 Success	
log	results.length+' countries found'
res	[{ "english_name": "Aruba" }, ...]

404 Could not find language/countries	
log	'Could not find countries for language: ' + lang
res	Could not find countries

Notes

None

Show info about country (showCountryInfo)

GET(http://localhost:3003/api/app/countries/info/:country_name)

country_name = english_name from countries

Request package

None

Responses

200 Success	
log	'Country found: '+country_name
res	[{ "country_iso_code": "AF", "english_name": "Afghanistan", "regions": "Western Asia", "intermediate_regions": "N/A", "continents": "Asia", "hdx_link": null, "hdx_description": null, "pt_link": null, "pt_description": null, "cg_link": "https://clearglobal.org/language-data-for-afghanistan/", "cg_description": null }]

404 Could not find country	
log	'Could not find country: '+country_name
res	Could not find country

Notes

None

Show all languages (showAllLang)

GET(<http://localhost:3003/api/app/lang/>)

Request package

None

Responses

200 Success	
log	results.length+' languages found'
res	[{ "lang_id": 0, "lang_name": "The testenese", "lang_status": "verified" }, ...]

Notes

None

Show language's details (showLangDetails)

GET(<http://localhost:3003/api/app/lang/:lang>)

lang = lang_name from languages

Request package

None

Responses

200 Success	
log	'Language found: '+lang
res	<pre>{ "language": { "lang_name": "Balochi", "iso_code": "bcc", "glottocode": "", "lang_status": "in progress" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "https://phoible.org/languages/sout2642", "description": "PHOIBLE" }, ...] }</pre>

404 Language not found	
log	'Could not find languages: '+lang
res	Could not find language

Notes

None

Show names of all languages (showAllLangNames)

GET(<http://localhost:3003/api/app/langNames/>)

Request package

None

Responses

200	Success
log	results.length+' languages found'
res	[{ "lang_name": "The testenese" }, ...]

Notes

None

Show all languages with details (showAllLangDetails)

GET(<http://localhost:3003/api/app/langDetails/>)

Request package

None

Responses

200 Success	
log	results.length+' languages found'
res	<pre>["Balochi": { "language": { "lang_name": "Balochi", "iso_code": "bcc", "lang_status": "in progress", "glottocode": "" }, "alternative_names": ["Balochi", ...], "links": [{ "link": "https://phoible.org/languages/sout2642", "description": "PHOIBLE" }, ...] }, ...]</pre>

Notes

None

Show language suggestions (showRequestsByLang)

GET(<http://localhost:3003/api/app/requests/:lang>)

lang = lang_name from languages

Request package

None

Responses

200 Success	
log	'Requests found, for language: '+lang
res	[{ "lang_request_id": 924, "created_user_id": 0, "assigned_user_id": 0, "lr_end_date": 0, "lr_start_date": 0, "lang_id": 0, "lr_type": "add", "lr_title": "Test", "lr_reason": "I need to test", "lr_lang_name": "Testenese", "lr_alternative_name": "Teston", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "complete" }, ...]

404 Requests not found	
log	'Could not find requests for language: '+lang
res	Could not find requests

Notes

None

Show language suggestions between dates (showRequestsBetweenDates)

GET(<http://localhost:3003/api/app/requests/period/:lang>)

lang = lang_name from languages

Request package

```
{
  "start_date":1682000000,
  "end_date":1682007000
}
```

Responses

200 Success

log 'Requests found, for language: '+lang

res [

```
  {
    "lang_request_id": 924,
    "created_user_id": 0,
    "assigned_user_id": 0,
    "lr_end_date": 0,
    "lr_start_date": 0,
    "lang_id": 0,
    "lr_type": "add",
    "lr_title": "Test",
    "lr_reason": "I need to test",
    "lr_lang_name": "Testenese",
    "lr_alternative_name": "Teston",
    "lr_iso_code": 0,
    "lr_glottocode": 0,
    "lr_added_countries": "",
    "lr_removed_countries": "",
    "lr_status": "complete"
  },
  ...
]
```

404 Requests not found

log 'Could not find requests for language: '+lang+'between these times: '+start_date+' and '+end_date

res Could not find requests

Notes

None

Show most recent suggestion (showMostRecentRequest)

GET(<http://localhost:3003/api/app/recent/>)

Request package

None

Responses

200	Success
log	Most recent request found
res	<pre>[{ "lang_request_id": 923, "created_user_id": 1561, "assigned_user_id": 0, "lr_end_date": 0, "lr_start_date": 1682007924, "lang_id": 0, "lr_type": "add", "lr_title": "Test", "lr_reason": "I need to test", "lr_lang_name": "Testenese", "lr_alternative_name": "", "lr_iso_code": 0, "lr_glottocode": 0, "lr_added_countries": "", "lr_removed_countries": "", "lr_status": "complete" }]</pre>

Notes

None

Generate token for password reset (forgotPassword)

POST(<http://localhost:3003/api/app/password/>)

Request package

```
{  
  "email": "testing@mail.com"  
}
```

Responses

200 Success	
log	'Link to reset password for user set. User email: '+email
res	{ "fieldCount": 0, "affectedRows": 1, "insertId": 0, "info": "", "serverStatus": 34, "warningStatus": 2 }

404 User not found	
log	{"code":"ER_BAD_NULL_ERROR","errno":1048,"level":"error","sql":"...", "sqlState":"23000"}
res	`User with email \${email} not found`

Notes

None

Extract token for password reset (extractToken)

GET(<http://localhost:3003/api/app/password/:email>)

email = email from users

Request package

None

Responses

200 Success	
log	'Token found. User id: '+email
res	[{ "reset_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyZXN1bHQiOnsiZWlhaWwiOiJ0ZXN0aW5nQG1haWwuY29tIn0sIm1hdCI6MTY4MjAyMTM4MiwiZXhwIjoxNjgyMDI0OTgyfQ.HaHsfsmRL4-Mat9GVrdX0_ua7R2PeIRB2EQpKQsP3fQ", "timestamp": 1682024982 }]

404 Token not found	
log	'Could not find token for user with email: ' + email
res	Could not find token

Notes

None
