Members

defaultController
Get default landing page aka default controller

errorController
Get a custom error-status page for non existing routes

UserInfo
After a User has been authenticated, the information of the user is stored in this object. The Content from this object is been transformed from a JWT Token.

authChecker ⇒ `UserInfo` | `undefined`
extracts the jwt from the keycloak context and transforms it to a UserInfo

Functions

initDB()
Truncates Database and injects Mock Database content for testing

~~*getUserInfo(req, _res, next)*~~
Decodes jwt token from request and appends UserInfo to request

jwtPayloadContentTransformer(payload) ⇒ `UserInfo`
Converts jwt payload to UserInfo

authChecker() ⇒ `Boolean`
checks if a Auth Token is present in the request and adds user to our database if not present

requestLogger(req, _res, next)
Middleware to Log all Express Requests

logAccess(info, next)
Middleware to Log all GraphQL Requests

paginate(list, paginated) ⇒ `Array.<T>`
Skip the first n elements and take the next n elements from the list

defaultController

Get default landing page aka default controller

Kind: global variable
Summary:

get function for defaultController

Access: public
Route: GET /
Author: LukasLJL

errorController

Get a custom error-status page for non existing routes

Kind: global variable
Summary:

get function for errorController

Access: public
Route: GET /abcd...
Author: LukasLJL

UserInfo

After a User has been authenticated, the information of the user is stored in this object. The Content from this object is been transformed from a JWT Token.

Kind: global variable
Summary:

UserInfo Object which Stores the information of the user

Author: LukasLJL

authChecker ⇒ `UserInfo` | `undefined`

extracts the jwt from the keycloak context and transforms it to a UserInfo

Kind: global variable

Summary:

get the UserInfo from the request

Returns: `UserInfo` | `undefined` -

- UserInfo

Author: LukasLJL

initDB()

Truncates Database and injects Mock Database content for testing

Kind: global function
Summary:

Initialise Database

Author: Jonas Eppard

~~getUserInfo(req, _res, next)~~

Deprecated

Decodes jwt token from request and appends UserInfo to request

Kind: global function
Summary:

Middleware to append decoded UserInfo to request

Author: Tim Horlacher

Param	Type	Description
req	`express.Request`	Express request object
_res	`express.Response`	Express response object

Param	Type	Description
next	`express.NextFunction`	Express next function

Example

```
api.get("/protected", keycloak.protect(), getUserInfo,
  (req: express.Request, res: express.Response) => {
    res.status(200);
    res.json("Hello " + req.UserInfo?.username);
  }
);
```

jwtPayloadContentTransformer(payload) ⇒ `UserInfo`

Converts jwt payload to UserInfo

Kind: global function
Summary:

Converts jwt payload to UserInfo

Returns: `UserInfo` -

- UserInfo

Author: Tim Horlacher

Param	Type	Description
payload	`jwt.JwtPayload`	jwt payload

authChecker() ⇒ `Boolean`

checks if a Auth Token is present in the request and adds user to our database if not present

Kind: global function
Summary:

authChecker middleware

Returns: `Boolean` -

- authenticated

Author: LukasLJL, Tim Horlacher

requestLogger(req, _res, next)

Middleware to Log all Express Requests

Kind: global function
Summary:

Middleware to Log all Express Requests

Author: LukasLJL

Param	Description
req	express request
_res	express response
next	express next function

logAccess(info, next)

Middleware to Log all GraphQL Requests

Kind: global function
Summary:

Middleware to Log all GraphQL Requests

Author: LukasLJL

Param	Description

Param	Description
info	graphql request
next	graphql next function

paginate(list, paginated) ⇒ `Array.<T>`

Skip the first n elements and take the next n elements from the list

Kind: global function
Summary:

Apply Pagination to a list

Returns: `Array.<T>` -
List of paginated elements

Author: Jonas Eppard

Param	Type	Description
list	`Array.<T>`	List to paginate
paginated	`PaginationArg`	Pagination Arguments skip and take