BBSN

Dashboard

Documentation technique

Aziz chigri
15/10/2018

Table des matières

1- Serveur

Documentation de l'api REST

Le / Les clients communiquent avec le serveur via le protocole http REST. Le serveur tourne sur le port 1908. Voici la liste des commandes générales que supporte l'API.

Titre	URL	Méthode	URL params	Header	Body	Succès	Echec
Se connecter	/login	POST	/	Content-type : application/json	{"username" : [string], "password" : [string]}	200 : OK `authorization:` Bearer <token>	403 : Forbidden
Créer User	/users/sign-up	POST	/	Content-type : application/json	{"username" : [string], "password" : [string], "lastName" : [string], "firstName" : [string], "email" : [string]}	200 : OK	400 : Bad Request
Mettre à jour User	/users/update	POST	/	Content-type : application/json `Authorization:` Bearer <token>	{"username" : [string], "password" : [string], "lastName" : [string], "firstName" : [string], "email" : [string]}	200 : OK	400 : Bad Request 403 : Forbidden
Récupérer les infos d'un user	/users	GET	username=[string]	Content-type : application/json `Authorization:` Bearer <token>	/	200 : OK {"username" : [string], "lastName" : [string], "firstName" : [string], "email" : [string]}	400 : Bad Request 403 : Forbidden

Stocker les préférences d'un user	/users/config	POST	/	Content-type : application/json
				Authorization:Bearer <token>
					{
					"username":[str],
					"widget":[
					{
					"name":[str]
					"size" :[str]
					"position":[str]
					"preference":[str]
					}
					]
					}
						200: OK
						Retourne le User
							400 : Bad Request
							403 : Forbidden

Récupérer la liste de tous les services	/services	GET	/	Content-type : application/json
				Authorization:Bearer <token>
					/	200 : OK
						[
						"name":
						"weather",
						"widgets": [
						{
						"name":
						"city_temperature",
						"description":
						"Affichage de la temperature pour une ville",
						"params": [
						{
						"name":
						"city",
						"type":
						"string"
						}
						],
						}
						],
						]
							403 : Forbidden

Récupérer la	/services/findOne	GET	serviceName=[string]	Content-type : application/json Authorization:Bearer <token>	/	{ "name": "weather",	403 :

liste de widgets d'un service en particulier						"widgets": [
						{
						"name": "city_temperature",
						"description": "Affichage de la temperature pour une ville",
						"params": [
						{
						"name": "city",
						"type": "string"
						}
						],
						}
						],}
							Forbidden

Récupérer la liste de tous les widgets	/widget	GET	/	Content-type : application/json
				Authorization:Bearer <token>
					/	200 : OK
						[
						{
						"name": "city_temperature",
						"description": "Affichage de la temperature pour une ville",
						"params": [
						{
						"name": "city",
						"type": "string"
						}
						],}]
							403 : Forbidden

2- Services choisis

Service	Nombre de Widgets	Documentation	Description
Météo	X	https://openweathermap.org/current	Affiche les données météo pour une ville donnée.
Taux de change	X	https://www.ecb.europa.eu/stats/policy_and_exchange_rates/euro_reference_exchange_rates/html/index.en.html (Dans for developers)	Affichage du taux de change d'un couple de monnaie EUR / M2
Steam	X	https://developer.valvesoftware.com/wiki/Steam_Web_API	Informations sur un jeu steam
Google	X	https://developers.google.com/gmail/api/	Mail non lu sur sa boite Gmail.
Google	X	https://cloud.google.com/translate/?hl=fr	Traduction d'un message dans une langue donnée
YouTube	X	https://developers.google.com/youtube/	Nombres abonnés sur sa chaine YT. Nombre de vue total sur la chaine

Pour requêter les services, tous les URL ont pour base **/services** Dans les sections suivantes, vous trouverez toutes les informations propres aux api REST de chacun des services.

a) Météo

Pour récupérer des informations sur le service météo :

URL	Méthode	URL params	Header	Body	Succès	Echec
/services/weather	POST	/	Content-type : application/json `Authorization:` Bearer <token>	{"city": [string]}	200 : OK	400 : bad request 403 : Forbidden

Voici un exemple de retour, si l'on requête l'api sur la ville de Toulouse :


```
{
   "coord":{
      "lon":1.44,
      "lat":43.6
   },
   "weather":[
      {
         "id":803,
         "main":"Clouds",
         "description":"broken clouds",
         "icon":"04n"
      }
   ],
   "base":"stations",
   "main":{
      "temp":290.15,
      "pressure":1015,
      "humidity":88,
      "temp_min":290.15,
      "temp_max":290.15
   },
   "visibility":10000,
   "wind":{
      "speed":1.5,
      "deg":210
   },
   "clouds":{
      "all":75
   },
   "dt":1539885600,
   "sys":{
      "type":1,
      "id":5535,
      "message":0.0021,
      "country":"FR",
      "sunrise":1539843166,
      "sunset":1539882303
   },
   "id":2972315,
   "name":"Toulouse",
   "cod":200
}
```

b) Taux de change

Pour récupérer des informations sur le service de taux de change, se référer au tableau ci-après. **Pour information, on ne peut pas positionner le filtre à date à plus de 90 jours avant la date d'émission de la requête.**

URL	Méthode	URL params	Header	Body	Succès	Echec
/services/exchange	GET	/	Content-type : application/json Authorization:Bearer <token>	/	200 : OK [Liste de toutes les monnaies prises en charge par l'api]	400 : bad request 403 : Forbidden
/services/exchange	POST	/	Content-type : application/json Authorization:Bearer <token>	{ "date":[string (yyyy-mm-dd)], "currency":[String]} **On peut initialiser un des deux champs à « vide » pour appliquer des filtres (voir exemples ci-dessous).**	200 : OK [Data] (ci-dessous)	400 : bad request 403 : Forbidden

Voici plusieurs exemples de retour, si l'on requête l'api sur une monnaie (USD) sans préciser de date:

```
{
   "rate":1.1505,
   "currency":"USD"
}
```

Si une date et une monnaie sont précisés (USD le 2018-10-15) :

```
{
   "rate":1.1581,
   "currency":"USD"
}
```

Si une date est précisée mais pas la monnaie (2018-10-15):

```
[
   {
      "rate":1.1581,
      "currency":"USD"
   },
   {
      "rate":129.53,
      "currency":"JPY"
   },
   {
      "rate":1.9558,
      "currency":"BGN"
   },
   {
      "rate":25.798,
      "currency":"CZK"
   },
   {
      "rate":7.4609,
      "currency":"DKK"
   },
   {
      "rate":0.88045,
      "currency":"GBP"
   },
   {
      "rate":323.45,
      "currency":"HUF"
   },
   {
      "rate":4.2942,
      "currency":"PLN"
   },
   {
      "rate":4.6673,
      "currency":"RON"
   },
   {
      "rate":10.392,
      "currency":"SEK"
   },
   {
      "rate":1.1428,
      "currency":"CHF"
   },
   {
      "rate":134.6,
      "currency":"ISK"
   },
   {
      "rate":9.4598,
      "currency":"NOK"
   },
   {
      "rate":7.4135,
      "currency":"HRK"
   },
   {
      "rate":75.924,
      "currency":"RUB"
   },
   {
      "rate":6.686,
      "currency":"TRY"
   },
   {
      "rate":1.6231,
      "currency":"AUD"
   },
   {
      "rate":4.3498,
      "currency":"BRL"
   },
{
      "rate":1.5085,
      "currency":"CAD"
   },
   {
      "rate":8.0135,
      "currency":"CNY"
   },
   {
      "rate":9.0767,
      "currency":"HKD"
   },
   {
      "rate":17614.7,
      "currency":"IDR"
   },
   {
      "rate":4.2011,
      "currency":"ILS"
   },
   {
      "rate":85.4735,
      "currency":"INR"
   },   {
      "rate":1307.61,
      "currency":"KRW"
   },
 {
      "rate":21.8192,
      "currency":"MXN"
   },
   {
      "rate":4.8131,
      "currency":"MYR"
   },
   {
      "rate":1.7723,
      "currency":"NZD"
   },
   {
      "rate":62.649,
      "currency":"PHP"
   },
   {
      "rate":1.5945,
      "currency":"SGD"
   },   {
      "rate":37.841,
      "currency":"THB"
   },
   {
      "rate":16.7129,
      "currency":"ZAR"
   }
]
```

c) Steam

Pour récupérer des informations sur le service steam :

URL	Méthode	URL params	Header	Body	Succès	Echec
/services/steam	GET	/	Content-type : application/json Authorization:Bearer <token>	/	200 : OK [Liste de tous les noms de jeux pris en charge par l'api] {"jeu":"appId"}	400 : bad request 403 : Forbidden
/services/steam	POST	/	Content-type : application/json Authorization:Bearer <token>	{"appId":[string]}	200 : OK [Data] (ci-dessous)	400 : bad request 403 : Forbidden

Voici plusieurs exemples de retours :

Si un GET est fait sur /services/steam le retour sera :


```
{
   "Factorio":"427520",
   "Life is Strange":"319630",
   "Undertale":"391540",
   "Clustertruck":"397950",
   "Tom Clancy's Splinter Cell: Chaos Theory":"13570",
   "PAYDAY 2":"218620",
   "Tom Clancy's Rainbow Six Siege":"359550",
   "Mirror's Edge":"17410",
   "Half-Life 2: Deathmatch":"320",
   "Assassin's Creed Odyssey":"812140",
   "Borderlands 2":"49520",
   "Just cause 3":"225540",
   "Just dance 2017":"446560",
   "NieR:Automata":"524220",
   "Portal 2":"620",
   "Dark Souls III":"374320",
   "Deus Ex: Mankind Divided":"337000",
   "Overcooked! 2":"728880",
   "Farming Simulator 19":"787860",
   "MONSTER HUNTER: WORLD":"582010"
}
```

Si on recherche les informations du jeu Factorio, le résultat sera :


```
{
   "appid":427520,
   "name":"Factorio",
   "developer":"Wube Software LTD.",
   "publisher":"Wube Software LTD.",
   "score_rank":99,
   "positive":38443,
   "negative":614,
   "userscore":98,
   "owners":"1,000,000 .. 2,000,000",
   "average_forever":4988,
   "average_2weeks":597,
   "median_forever":1972,
   "median_2weeks":220,
   "price":"3000",
   "initialprice":"3000",
   "discount":"0",
   "languages":"English, French, Italian,
German, Spanish - Spain, Hungarian, Dutch,
Norwegian, Polish, Portuguese, Portuguese –
Brazil, Romanian, Finnish, Swedish, Czech,
Russian, Ukrainian, Japanese, Simplified Chinese,
 Traditional Chinese, Korean",
   "genre":"Casual, Indie, Simulation, Strategy,
Early Access",
   "ccu":8874,
   "tags":{
      "Early Access":480,
      "Base-Building":1357,
      "Resource Management":1199,
      "Sandbox":1035,
      "Crafting":959,
      "Strategy":898,
      "Survival":785,
      "Multiplayer":745,
      "Open World":693,
      "Management":671,
      "Co-op":490,
      "Singleplayer":488,
      "Simulation":479,
      "Moddable":452,
      "Trains":425,
      "Indie":411,
      "Sci-fi":373,
      "Aliens":353,
      "Tower Defense":263
   }
}
```

d) Spotify

Pour récupérer des informations sur le service Spotify, se référer au tableau ci-après.

URL	Méthode	URL params	Header	Body	Succès	Echec
/services/spotify/filter	GET	/	Content-type : application/json Authorization:Bearer <token>	/	200 : OK [Liste de tous les marchés + types pris en compte par l'api de recherche] (Voir exemples ci-dessous)	400 : bad request 403 : Forbidden
/services/spotify/search	POST	/	Content-type : application/json Authorization:Bearer <token>	{"**keyword**":[string], "**type**":[string], "market":[string], "limit":[string]} **Les champs en gras sont obligatoires.**	200 : OK [Data] (link)	400 : bad request 403 : Forbidden

Si on veut récupérer la liste de filtres (GET sur /spotify/filter) :


```
{
   "market":[
      "AD",
      "AR",
      "AT",
      "AU",
      "BE",
      "BG",
      "BO",
      "BR",
      "CA",
      "CH",
      "CL",
      "CO",
      "GR",
      "GT",
      "HK",
      "HN",
      "HU",
      "ID",
      "IE",
      "IL",
      "IS",
      "IT",
      "JP",
      "LI",
      "LT",
      "LU",
      "PL",
      "PT",
      "PY",
      "RO",
      "SE",
      "SG",
      "SK",
      "SV",
      "TH",
      "TR",
      "TW",
      "US",
      "UY",
      "VN",
```

```
        "CR",
        "CY",
        "CZ",
        "DE",
        "DK",
        "DO",
        "EC",
        "EE",
        "ES",
        "FI",
        "FR",
        "GB",
        "LV",
        "MC",
        "MT",
        "MX",
        "MY",
        "NI",
        "NL",
        "NO",
        "NZ",
        "PA",
        "PE",
        "PH",
        "ZA"
    ],
    "type":[
        "album",
        "artist",
        "playlist",
        "track"
    ]
}
```

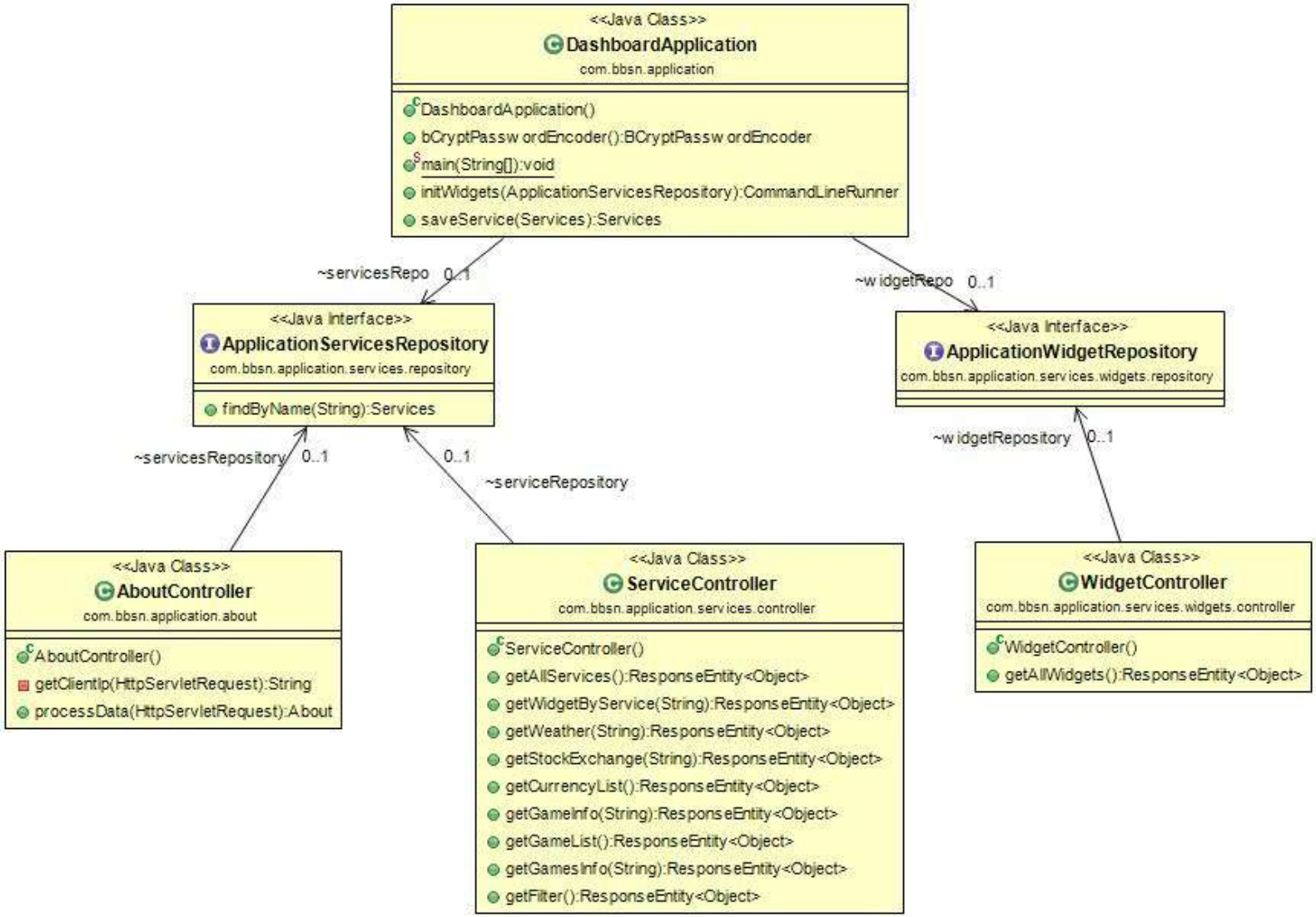

3- Widgets

Service	Widget	Parametres	Donné affiché	Description
Weather	City temperature	-city	-City -Température -Description	Affiche la température pour une ville donnée.
	City weather	-city	-City -Température -Description -Taux d'humidité -Vitesse du vent -Direction du vent -Pression atmosphérique	Affiche les données météo plus précises pour une ville donnée.
Exchange rate	Exchange	- currency	-Taux de change	Affichage du taux de change d'un couple de monnaie EUR / M2 à l'instant T.
Steam	Game information	- appId	-Name -App ID -Développer -Prix en dollar -Discount	Informations basiques sur un jeu Steam

	Game statistics	- appId	-Name -App ID -Classement steam -Nombre avis positifs -Nombre avis négatifs -Moyenne des avis (en pourcentage)	Affiche les statistiques sur un jeu Steam.
Spotify	Track_list	- Album - Artist - Playlist	- Nom des musiques	Affiche la liste des musiques de l'album / artiste / playlist.
	Music_info	- album - artist - track	- Nombre de musiques (si album / artiste) - durée (si album / musique) - Titre / nom - Popularité (note sur 100) - Date de sortie (si album /.musique)	Affiche les informations sur l'album / l'artiste / la musique

4) Architecture

Le diagramme de classe suivant représente l'association des classes permettant l'interaction avec la BDD (Repository) et celles permettant l'interaction avec le client via REST (Controller).



Le diagramme de classe suivant représente la manière dont sont stockés les services / widgets.