

ESP-Renty

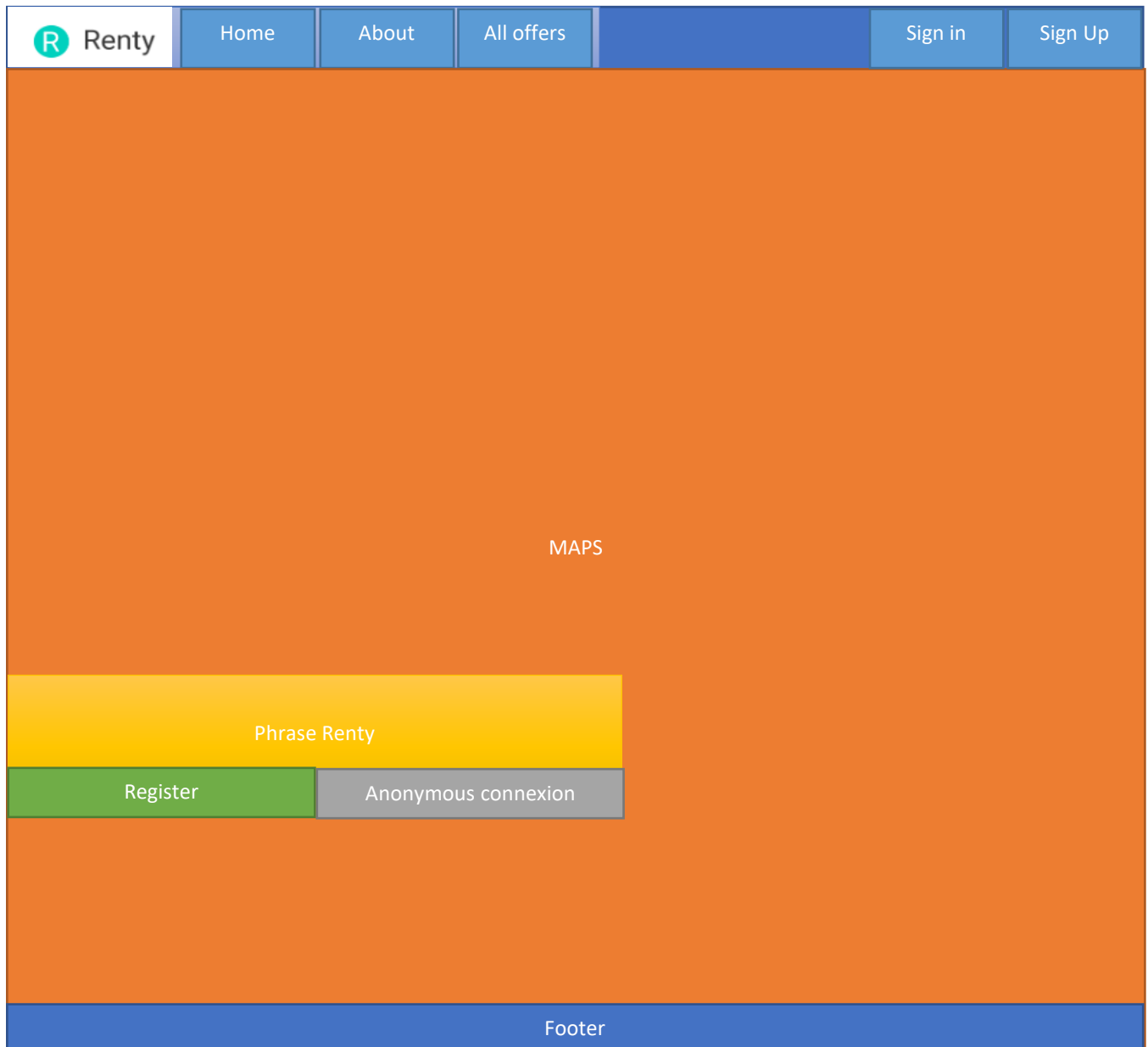
Document Fonctionnalité

Nom / Prénom	Modification	Date
Guillaume RAULT	Création document	21 / 09 / 2020

Table des matières

ESP-Renty	1
Connection Back-Front	4
Query	4
Mutations.....	6
Routes – front	7
Routes - Back	11
Sous-Type.....	14
Enum	16

Mock-Up Générique



hom

Connection Back-Front

Query

Back	Front
Offers	RentOffers : ⇒ OffersList.svelte ⇒ Map.svelte
Login	LOGIN : ⇒ login.svelte
checkEmail	None
CheckUsername	None
SocialLogin	SOCIALLOGIN : ⇒ login.svelte
User	User : ⇒ UserProfile.svelte
Add2FA	None
myoffers	None
mydocuments	MyDocuments : ⇒ DocList.svelte
confirm	CONFIRM : ⇒ login.svelte
likeOffer	None
notify	None
userbyname	None

checkmein	None
------------------	------

Mutations

Back	Front
addOffer	None
register	REGISTER : ⇒ Register.svelte
socialregister	SOCIALREGISTER : ⇒ Register.svelte
uploadasset	UploadAsset : ⇒ DocUpload.svelte
deleteasset	None
updateasset	None
updatePassword	None
updatecontact	UPDATECONTACT : ⇒ UpdateProfile.svelte

Routes – front

Documentation	Argument
REGISTER	<pre>mutation(\$name: String! \$password: String! \$email: String \$notificationID: String) { register(input: { name: \$name password: \$password email: \$email notificationID: \$notificationID }) { name } }</pre>
Login	<pre>query(\$name: String!, \$password: String!, \$notificationID: String) { login(input: { name: \$name password: \$password notificationID: \$notificationID }) { token otp } }</pre>

SOCIALLOGIN

```
query($IDToken: String!, $notificationID: String) {
  socialLogin(
    input: {
      IDToken: $IDToken
      provider: GOOGLE
      notificationID: $notificationID
    }
  ){
    token
    otp
  }
}
```

SOCIALREGISTER

```
mutation($IDToken: String!, $notificationID: String) {
  socialregister(
    input: {
      IDToken: $IDToken
      provider: GOOGLE
      notificationID: $notificationID
    }
  ){
    name
  }
}
```

MyDocuments

```
query {
  mydocuments {
    label {
      text
    }
    asset {
      type
      name
      url
    }
    doctype
    createdAt
    updatedAt
  }
}
```

UploadAsset

```
mutation($documentType: DocumentType!, $label: String!, $file: Upload!) {
  uploadasset(
    input: { documentType: $documentType, label: $label, file: $file }
  )
}
```


RentOffers	<pre>query { offers { title { text } location { latitude longitude } } }</pre>
RentOffers	<pre>query { offers { area { unit value } price { curr value } title { text } assets { url } } }</pre>
UPDATECONTACT	<pre>mutation(\$firstName: String \$lastName: String \$civility: Civility \$email: String \$phone: PhoneInput) { updatecontact(input: { firstName: \$firstName lastName: \$lastName civility: \$civility email: \$email phone: \$phone }) { name</pre>

	<pre>} }</pre>
USER	<pre>query { user { name civility firstName lastName preferredLangage email { value verified } phone { value verified } createdAt updatedAt } }</pre>

Routes - Back

Documentation	Mise en place Front	Argument
Offers		offers: [RentOffer !] ! <pre> type RentOffer { id: String area: [Area!]! price: [Price!]! title: [Label!]! assets: [Asset!]! location: Location! description: [Label!]! availableFrom: String! createdAt: String! updatedAt: String! } </pre>
login	Fonctionnel Const LOGIN	login(input: UserLoginInput!): Auth <pre> type Auth { token: String! type: String! otp: Boolean! createdAt: String! validity: String! } </pre>
checkEmail		checkEmail(email: String!): Boolean!
checkUsername		checkUsername(name: String!): Boolean!
socialLogin		socialLogin(input: SocialLoginInput!): Auth <pre> type Auth { token: String! type: String! } </pre>

		otp: Boolean! createdAt: String! validity: String! }
User		user: User! type User { name: String! firstName: String lastName: String civility: Civility! preferredLangage: LangageCode! likedOffers: [RentOffer!] email: Email phone: PhoneNumber devices: [Device!] createdAt: String! updatedAt: String! }
Add2FA		add2FA: Totp type Totp { url: String! qrcode: String! }
Myoffers		myoffers: [RentOffer!]! type RentOffer { id: String area: Area! price: [Price!]! title: [Label!]! assets: [Asset!]! location: Location! description: [Label!]! availableFrom: String! createdAt: String! updatedAt: String! }
Mydocuments		mydocuments: [UserDocument!]! type UserDocument { label: Label! asset: Asset! doctype: DocumentType! createdAt: String! updatedAt: String! }

Confirm		confirm (code: Int!): Auth type Auth { token: String! type: String! otp: Boolean! createdAt: String! validity: String! }
likeOffer		likeOffer (offerId: String! value: Boolean!): Boolean!
notify		type NotifyReport { sent: Int! devices: [DeviceType!]! } type NotifyReport { sent: Int! devices: [DeviceType!]! }
userbyname		userbyname (name: String!): User type User { name: String! firstName: String lastName: String civility: Civility! preferedLangage: LangugeCode! likedOffers: [RentOffer!] email: Email phone: PhoneNumber devices: [Device!] createdAt: String! updatedAt: String! }
checkmein		checkmein (offer: String!

): Boolean
--	--	-------------------

Sous-Type

Nom	Type
Area	area: Area ! <pre>type Area { unit: AreaUnit! value: Float! updatedAt: String! }</pre>
Price	price: [Price !] ! <pre>type Price { curr: Currency! value: Float! createdAt: String! updatedAt: String! }</pre>
title	title: [Label !] ! <pre>type Label { text: String! lang: LangageCode! createdAt: String! updatedAt: String! }</pre>
assets	assets: [Asset !] ! <pre>type Asset { url: String! name: String! type: ContentType! createdAt: String! updatedAt: String! }</pre>
location	location: Location ! <pre>type Location { latitude: Float!</pre>

	<pre>longitude: Float! metadatas: String }</pre>
description	<pre>description: [Label] type Label { text: String! lang: LangageCode! createdAt: String! updatedAt: String! }</pre>
Input	<pre>input: UserLoginInput! type UserLoginInput { name: String email: String phone: String password: String! notificationID: String }</pre>

Enum

Enum name	Argument
Lang	lang: LangageCode enum LangageCode { EN FR ES IT PL }
type	type: ContentType! enum ContentType { PHOTO VIDEO DOCUMENT }
Curr	curr: Currency enum Currency { EUR USD BTC }
Devices	devices: [DeviceType!] enum DeviceType { DESKTOP MOBILE TABLET OTHER }