## REACT

Dans le cadre de votre entretien, vous devez effectuer un petit exercice sur REACT.

Vous devrez créer une simple page listant les héros Marvel et une page de détail comme sur les écrans suivants :

# Liste des super héros:



















### Fiche identité:



### A.I.M.

AIM is a terrorist organization bent on destroying the world.

### Comics

Ant-Man & the Wasp (2010) #3 Avengers (1998) #67

Avengers (1963) #87

Avengers Vol. 2: Red Zone (Trade Paperback)

Avengers Vol. II: Red Zone (Trade Paperback)

Captain America (1998) #28

Defenders (1972) #57

Incredible Hulks (2009) #606 (VARIANT)

Marvel Adventures Super Heroes (2010) #16

Marvel Masterworks: Captain America Vol. (Hardcover)

Marvel Masterworks: Captain America Vol. 1 - 2nd Edition (Hardcover)

Marvel Masterworks: Doctor Strange Vol. (Hardcover)

Marvel Masterworks: Doctor Strange Vol. (Hardcover)
Marvel Masterworks: The Invincible Iron Man Vol. (Hardcover)

Marvel Masterworks: The Invincible Iron Man Vol. 1 (Hardcover)

Marvel Masterworks: The Silver Surfer Vol. 2 (Hardcover)

Strange Tales (1951) #146

Strange Tales (1951) #147

Strange Tales (1951) #148

Tales of Suspense (1959) #79

#### Serie

Ant-Man & the Wasp (2010 - 2011)

Avengers (1963 - 1996)

Avengers (1998 - 2004)

Avengers Vol. 2: Red Zone (2003)

Avengers Vol. II: Red Zone (2003)

Captain America (1998 - 2002)

Defenders (1972 - 1986) Incredible Hulks (2009 - 2011)

Marvel Adventures Super Heroes (2010 - 2012)

Marvel Masterworks: Captain America Vol. (2005)

Marvel Masterworks: Captain America Vol. 1 - 2nd Edition (2003)

Marvel Masterworks: Doctor Strange Vol. (2005)

Marvel Masterworks: The Invincible Iron Man Vol. (2005)

 ${\sf Marvel\ Masterworks: The\ Invincible\ Iron\ Man\ Vol.\ 1}\ (0000\ -\ Present)$ 

Marvel Masterworks: The Silver Surfer Vol. 2 (2003) Strange Tales (1951 - 1968)

Tales of Suspense (1959 - 1968)

Tales of Suspense Transport (1959 - 1968)



## Côté technique:

Pour ce faire voici les éléments API nécessaires aux appels :

- API\_PUBLIC: 298bab46381a6daaaee19aa5c8cafea5

- API\_PRIVATE: b0223681fced28de0fe97e6b9cd091dd36a5b71d

- BASE\_URL: http://gateway.marvel.com:80

Pour les appels aux données voici les endpoints :

- Characters :
  - o Method: GET
  - Uri : /vl/public/characters
  - o queryParams:
    - **ts** -> timestamp au moment de l'appel en seconde
    - apikey -> API\_PUBLIC
    - hash -> concaténation du timestamp, clef privée, clef public hashé en md5
       e.g: en utilisant crypto par exemple:
       crypto.createHash('md5').update('{concatenatedString}').digest('hex')
- Characters Info :
  - o Method: GET
  - o Uri :/vl/public/characters/{characterId}
  - o queryParams:
    - ts -> timestamp au moment de l'appel en seconde
    - apikey -> API\_PUBLIC
    - hash -> concaténation du timestamp, clef privée, clef public hashé en md5
       e.g: en utilisant crypto par exemple:
       crypto.createHash('md5').update('{concatenatedString}').digest('hex')

Vous êtes libre dans l'utilisation de librairie, router, fetching... Un plus serait une implémentation d'un pattern flux.

Vous n'êtes pas tenu de respecter la charte graphique des écrans ci-dessus.

Vous devrez nous livrer le code source de cet exercice.

Merci & bon react

