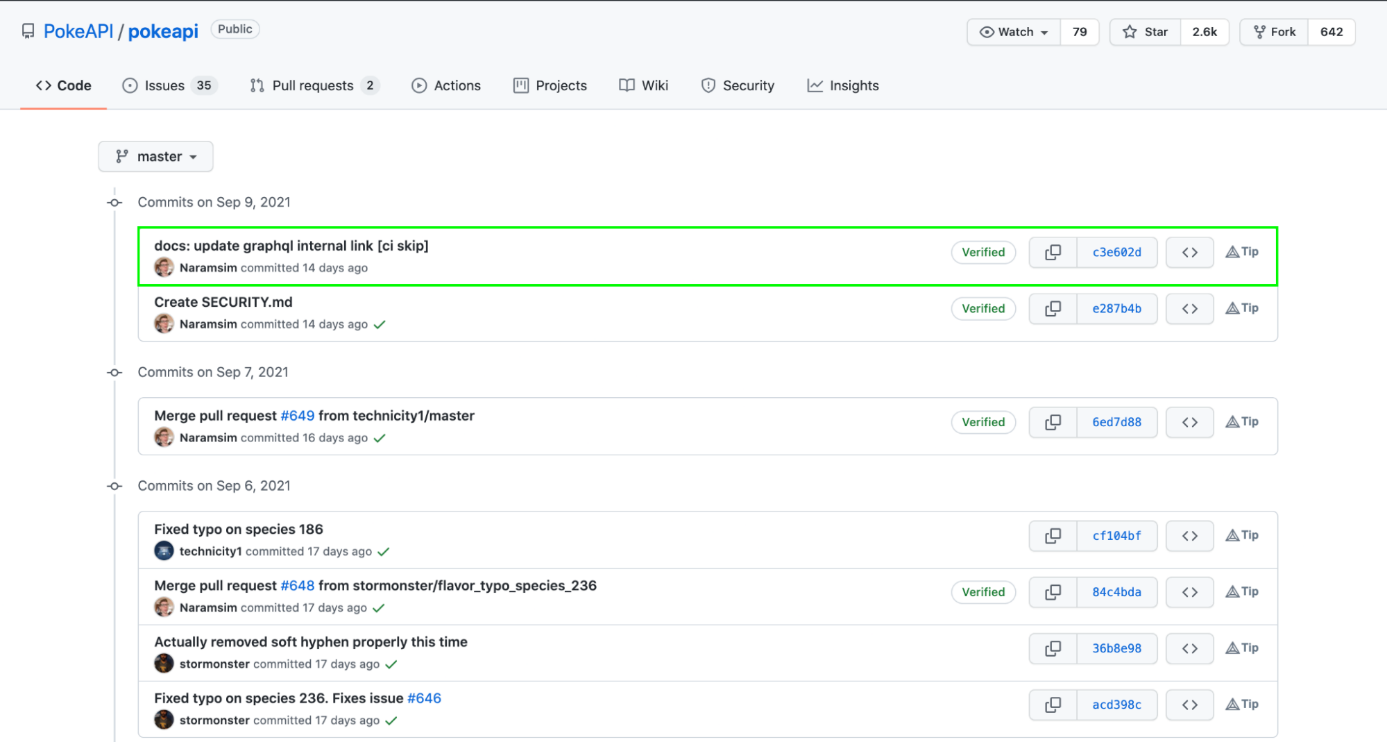
* Toutes les API ne sont pas RESTful et les API REST ont des lignes directrices architecturales spécifiques.
* Les avantages clés des API REST sont les suivants :
  + la séparation du client et du serveur, qui aide à scaler plus facilement les applications ;
  + le fait d’être stateless, ce qui rend les requêtes API très spécifiques et orientées vers le détail ;
  + la possibilité de mise en cache, qui permet aux clients de sauvegarder les données, et donc de ne pas devoir constamment faire des requêtes aux serveurs.
* SOAP est un autre type d’API, mais est plus utilisé dans les grandes entreprises.

Vous venez de voir la structure d’une API REST et ses avantages ; il est temps de voir ce qui constitue une API REST : les ressources. Suivez-moi dans le prochain chapitre, et attaquons-nous aux ressources !

* Une ressource est un objet de type nominal utilisé pour sauvegarder des données dans une API.
* Une ressource peut contenir des informations supplémentaires.
* Les ressources sont regroupées en collection et sont nommées au pluriel.
* Vous pouvez accéder aux ressources dans les API avec des URI.
* Les données REST peuvent être en langage JSON ou XML, mais le JSON est le plus courant.
* CRUD signifie create (créer), read (lire), update (mettre à jour) et delete (supprimer).
* GET est le verbe HTTP pour obtenir des données, et il est généralement utilisé avec un ID pour obtenir une donnée spécifique.
* Les applications utilisent GET pour présenter des informations sur des pages web.
* Utilisez Postman pour tester les API.
* La documentation est le manuel d’utilisation d’une API.
* La documentation vous permet de trouver la liste des endpoints accompagnée du verbe HTTP correspondant.

Un exemple d’une API mise à jour récemment #PokeApi

Vous pouvez constater sur leur GitHub que la date de leur dernier commit est assez récente ; vous savez donc que cette API est maintenue et mise à jour. 😉

Avant chaque utilisation d’une API externe, vérifiez la date de la dernière mise à jour sur GitHub ou bien sur lson site Internet, lisez la documentation et si vous avez des doutes (ou non), regardez en ligne des avis ou posez simplement la question à un autre développeur.

**En résumé**

* Pour obtenir un token d’authentification GitHub, vous pouvez :
  + aller sur <https://github.com/settings/developers> et cliquer sur **Personal Access Tokens**(tokens d’accès personnel) ;
  + cliquer sur **Generate new token** (générer nouveau token) et saisir « Open Classrooms » dans les notes ;
  + vous verrez alors toutes les différentes options d’autorisation dont vous voulez doter votre API ;
  + choisissez toutes celles que vous voulez tester. Nous avons utilisé Repo et Delete (supprimer) ;
  + cliquez sur **Generate token**(générer token) et obtenez votre token API personnel ; 👏
  + assurez-vous de copier et sauvegarder ce token, car nous l’utiliserons par la suite.
* L’authentification est nécessaire pour garantir que seules les personnes avec les *autorisations adéquates* peuvent accéder à votre API.
* Les clés ou tokens API sont couramment utilisés dans une requête pour authentifier un utilisateur.
* Assurez-vous de bien vérifier la fiabilité d’une API avant de l’utiliser.
* Vous pouvez trouver les indications de mise à jour d’une API sur GitHub ou sur son site Internet.