

Bonnes pratiques de programmation en Python

Groupe Informatique - IREM Bordeaux

Corentin LATAPPY - Mercredi 24 Mai 2023

Corentin Latappy

 corentin.latappy@labri.fr

Corentin Latappy

✉ corentin.latappy@labri.fr

Thèse CIFRE au **LaBRI** en collaboration avec  **promyze**

Corentin Latappy

✉ corentin.latappy@labri.fr

Thèse CIFRE au **LaBRI** en collaboration avec  **promyze**

Sujet autour de l'apprentissage automatique des pratiques de code

Corentin Latappy

✉ corentin.latappy@labri.fr

Thèse CIFRE au **LaBRI** en collaboration avec  **promyze**

Sujet autour de l'apprentissage automatique des pratiques de code

Startup éditrice d'un outil destiné à favoriser l'échange de pratiques entre développeurs

Pratiques de code

Quoi est-ce ?

Pratiques de code

Quoi est-ce ?

Ensemble de règles à suivre

Pratiques de code

Quoi est-ce ?

Ensemble de règles à suivre

Function and Variable Names

Function names should be lowercase, with words separated by underscores as necessary to improve readability.

Variable names follow the same convention as function names.

mixedCase is allowed only in contexts where that's already the prevailing style (e.g. `threading.py`), to retain backwards compatibility.

<https://peps.python.org/pep-0008/#function-and-variable-names>

Pratiques de code

Pourquoi ?

Pratiques de code

Pourquoi ?

Contribuent et améliorent la qualité logicielle

Pratiques de code

Pourquoi ?

Contribuent et améliorent la qualité logicielle

- Lisibilité

Pratiques de code

Pourquoi ?

Contribuent et améliorent

- Lisibilité



```
1  # Bad
2  def long_function_name(
3      var_one, var_two, var_three,
4      var_four):
5      print(var_one)
6
7  # Good
8  def long_function_name(
9      var_one, var_two, var_three,
10     var_four):
11     print(var_one)
```

Pratiques de code

Pourquoi ?

Contribuent et améliorent la qualité logicielle

- Lisibilité
- Cohérence

Pratiques de code

Pourquoi ?

Contribuent et améliorent

- Lisibilité
- Cohérence



```
1  def sayHello(name):
2      formula = 'Hello'
3      print(formula + " " + name + '!')
4
5  def say_good_bye(name):
6      formula = "Good bye"
7      print(formula + ' ' + name)
8
9  sayHello('IREM')
10 say_good_bye("IREM")
```

Pratiques de code

Pourquoi ?

Contribuent et améliorent la qualité logicielle

- Lisibilité
- Cohérence
- Maintenabilité

Pratiques de code

Choix

Pratiques de code

Choix

- Littérature
 - Clean Code, *Robert C. Martin*

Pratiques de code

Choix

- Littérature
 - Clean Code, *Robert C. Martin*
- Conventions, Documentations
 - PEP 8 – Style Guide for Python Code

Pratiques de code

Choix

- Littérature
 - Clean Code, *Robert C. Martin*
- Conventions, Documentations
 - PEP 8 – Style Guide for Python Code
- Expériences, Échanges

Linters et Outils d'Analyse Statique

Linters et Outils d'Analyse Statique

Détection automatique et intégrable dans une CI

Linters et Outils d'Analyse Statique

Détection automatique et intégrable dans une CI

Intégré directement dans les IDEs

Linters et Outils d'Analyse Statique

Détection automatique et intégrable dans une CI

Intégré directement dans les IDEs

Correction automatique

Linters et Outils d'Analyse Statique

Démo avec Pylint sur VSCode

Linters et Outils d'Analyse Statique

Linters et Outils d'Analyse Statique

Peuvent produire des « faux positifs »

Linters et Outils d'Analyse Statique

Peuvent produire des « faux positifs »

Sont limités aux règles proposées par l'éditeur de l'outil

Linters et Outils d'Analyse Statique

Peuvent produire des « faux positifs »

Sont limités aux règles proposées par l'éditeur de l'outil

Problématique avec les pratiques « maisons »

Linters et Outils d'Analyse Statique

Peuvent produire des « faux positifs »

Sont limités aux règles proposées par l'éditeur de l'outil

Problématique avec les pratiques « maisons »

Ne permettent pas de détecter toutes les règles

Code Reviews

Code Reviews

Systeme de revue par des pairs

Code Reviews

Systeme de revue par des pairs

Permet d'assurer la diffusion et de vérifier l'application des pratiques

Code Reviews

Systeme de revue par des pairs

Permet d'assurer la diffusion et de vérifier l'application des pratiques

Nécessite l'expertise humaine

Code Reviews

Systeme de revue par des pairs

Permet d'assurer la diffusion et de vérifier l'application des pratiques

Nécessite l'expertise humaine

Échange souvent unilatéral et qui ne profite pas à tout le monde

Promyze - Ateliers Crafts

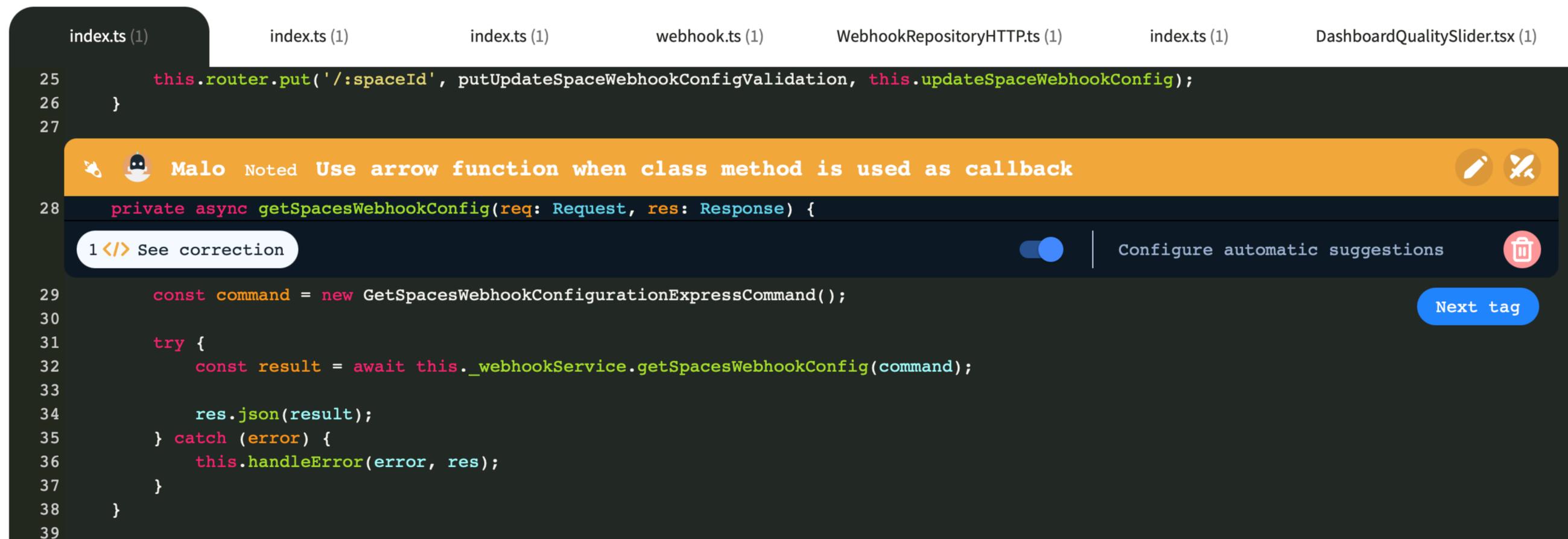
Promyze - Ateliers Crafts

Création de pratique depuis les IDEs

Promyze - Ateliers Crafts

Création de pratique depuis les IDEs

Temps d'échange entre les développeurs



The screenshot shows an IDE interface with several tabs at the top: 'index.ts (1)', 'index.ts (1)', 'index.ts (1)', 'webhook.ts (1)', 'WebhookRepositoryHTTP.ts (1)', 'index.ts (1)', and 'DashboardQualitySlider.tsx (1)'. The active tab is 'index.ts (1)'. The code editor displays the following TypeScript code:

```
25     this.router.put('/:spaceId', putUpdateSpaceWebhookConfigValidation, this.updateSpaceWebhookConfig);
26   }
27
28   private async getSpacesWebhookConfig(req: Request, res: Response) {
29     const command = new GetSpacesWebhookConfigurationExpressCommand();
30
31     try {
32       const result = await this._webhookService.getSpacesWebhookConfig(command);
33
34       res.json(result);
35     } catch (error) {
36       this.handleError(error, res);
37     }
38   }
39
```

A linting error notification is displayed in an orange bar above the code editor. The notification reads: "Malo Noted Use arrow function when class method is used as callback". To the right of the notification are icons for edit and delete. Below the notification, there is a button labeled "1 </> See correction" and a toggle switch for "Configure automatic suggestions". A "Next tag" button is also visible on the right side of the code editor.

Pratiques de code

Et la Recherche dans tout ça ?

Pratiques de code

Et la Recherche dans tout ça ?

Apprentissage des pratiques

Pratiques de code

Et la Recherche dans tout ça ?

Apprentissage des pratiques

Correction automatique

Pratiques de code

Rien n'est gravé dans le marbre

Bonnes pratiques de programmation en Python

Groupe Informatique - IREM Bordeaux

Corentin LATAPPY - Mercredi 24 Mai 2023