

# COVLint

Coverity Connectサーバーの特長の一つである問題のある行に指摘が付いたソースコード画面をVisual Studio Code上に実現しました。Language Server Protocol (言語サーバー) を利用していますので、言語サーバーの機能があるAtom、VimやEmacsなどのエディターで動作します（とされていますが未確認です）。

- Coverity Connect ソースコード画面

CID	ファイル	機能	言語	影響度	行...	問題の種類 ▲	タイプ
442930	/home/coverity/...	add_use	Python 3	中	399	セキュリティ	SSL 証明書の検証が無効になっています
442931	/home/coverity/...	get_grou	Python 3	中	267	セキュリティ	SSL 証明書の検証が無効になっています
442932	/home/coverity/...	cov_get_	Python 3	中	1...	セキュリティ	SSL 証明書の検証が無効になっています
442933	/home/coverity/...	delete_u	Python 3	中	499	セキュリティ	SSL 証明書の検証が無効になっています
442934	/home/coverity/...	cov_posi	Python 3	中	1...	セキュリティ	SSL 証明書の検証が無効になっています
442935	/home/coverity/...	get_total	Python 3	中	128	セキュリティ	SSL 証明書の検証が無効になっています
442936	/home/coverity/...	cov_posi	Python 3	中	2...	セキュリティ	SSL 証明書の検証が無効になっています
442937	/home/coverity/...	cov_posi	Python 3	中	2...	セキュリティ	SSL 証明書の検証が無効になっています

21 件中 1 件の 問題 : スナップショット別 を選択

gitlablib.py

```
392 url = url_prefix + "groups/" + str(group_id) + "/members"
393 headers = {"Private-Token": private_token}
394 response = requests.post(
395     url,
396     proxies=proxies_dic,
397     headers=headers,
398     data={"user_id": user_id, "access_level": level},
399     verify=False,
400 )
401
```

❖ CID 442930: (#1 of 1): SSL 証明書の検証が無効になっています (BAD\_CERT\_VERIFICATION)

1. ssl\_verify\_disabled: パラメータ verify は requests.post 関数で明示的に False に設定されています。

💡 verify を True に設定するか、デフォルト値が True であるため省略します。

- ```

383     group_id = group_info[2]["ITS"]
384 elif group == "TIC":
385     group_name = "TIC"
386     group_id = group_info[3]["TIC"]
387 else:
388     group_name = "None"
389     group_id = 0
390 """ You, 13 か月前 • Power Automate テスト用
391
392 url = url_prefix + "groups/" + str(group_id) + "/members"
393 headers = {"Private-Token": private_token}
394 res = requests.get(url, headers=headers)
395
396 CID: 442930: SSL 証明書の検証が無効になっています (BAD_CERT_VERIFICATION)
397 ssl_verify_disabled: パラメータ verify は requests.post 関数で明示的に False に設定されています。
398 covlint
399
400 verify=False,
401
402 # 200/201:登録成功、409:既に登録済みのため例外には投げる
403 if (
404     response.status_code != 200
405     and response.status_code != 201
406     and response.status_code != 409
407     and response.status_code

```

- ```

extension > cov_lint > !azure-20217-linkage-kombis > /template.yaml > {} resources > {} table > {} Properties > {} !tablename
109 # レイヤー
110 CommonLayer:
111   CID: 439204: ストレージ パージョンの無効化 (SIGMA.storage_versioning_disabled)
112   Sigma main event: Amazon DynamoDB テーブルで、ポイントインタイム リカバリー (PITR) が有効になっていません。
113   その結果、誤って書き込みや削除を実行した場合の保護がありません。 covlint
114   Pr
115   CID: 439197: 顧客入力暗号化キーの欠如 (SIGMA.vendor_provided_encryption_key)
116   Sigma main event: Amazon DynamoDBサービスは、お客様が管理するKMSキーを使用せず、AWSが所有するキーまたはAWSが
117   管理するKMSキーを使用してデータを暗号化するように構成されています。 covlint
118   CID: 439190: データベースの暗号化が無効です (SIGMA.database_encryption_disabled)
119   Sigma main event: Amazon DynamoDB テーブルでサーバー側の暗号化が有効になっていないため、保存中のデータが露出
120   しています。 covlint
121 # 連
122 Tab
123 Ty
124   Unresolved tag: !Sub YAML
125   問題の表示 (Alt+F8) クイックフィックス... (Ctrl+.)
126   複数の指摘を表示可能
127   TableName: !Sub ${Environment} region equipment-linkage-table
128   AttributeDefinitions:
129     - AttributeName: uid
130     - AttributeType: S
131     - AttributeName: linkage-uid

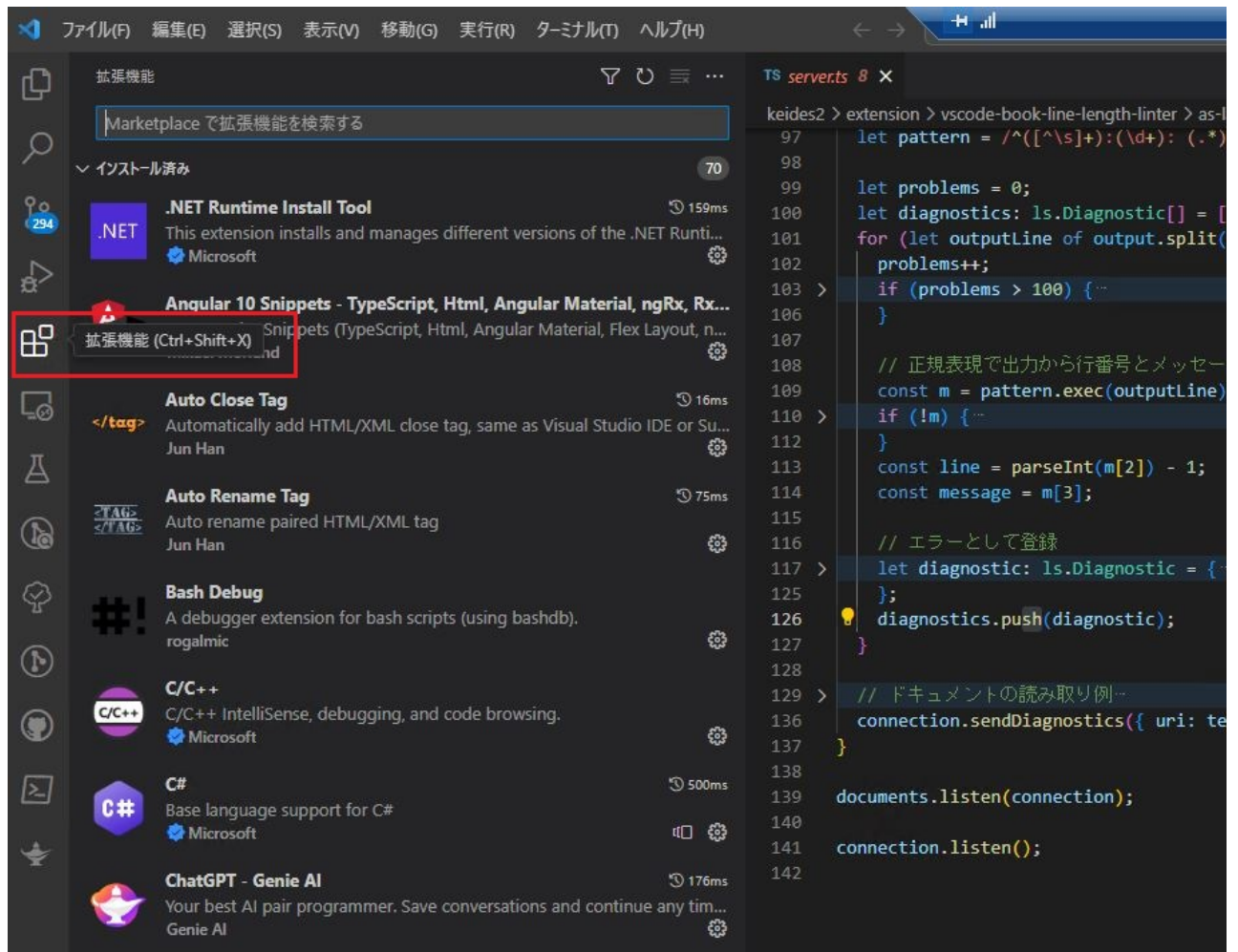
```

指摘結果は、スクリプト `cov_snap` が取得した CSV ファイルから読み込みます。

## 実施手順

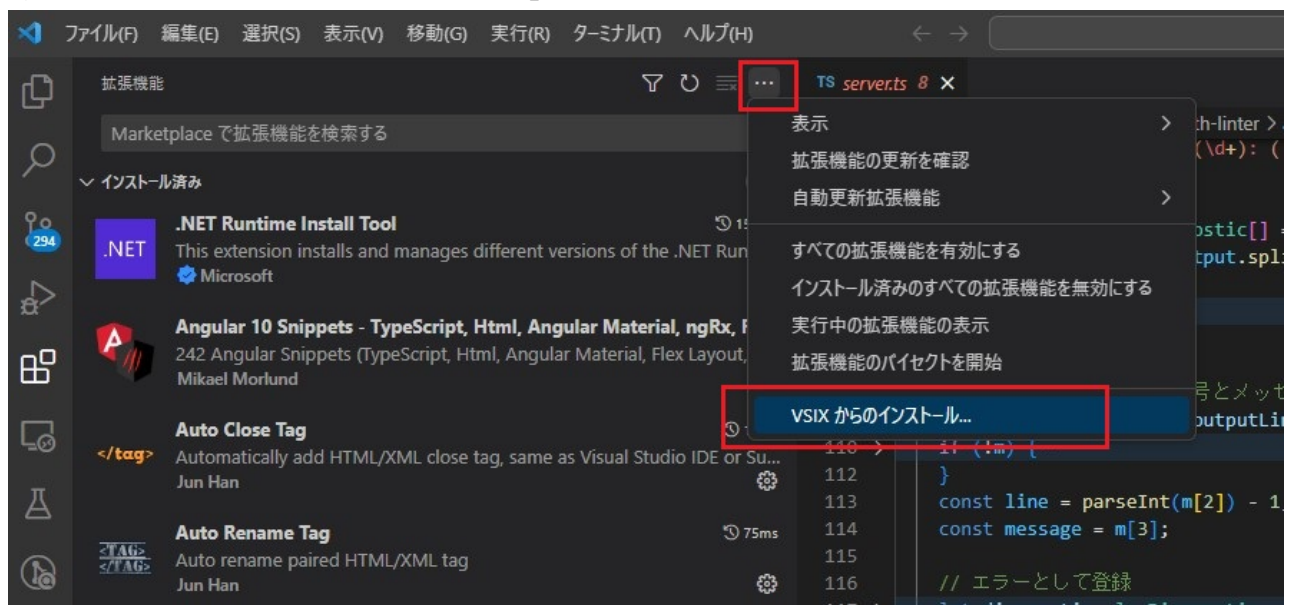
COVLintは、VSIXパッケージファイルからインストールする方法と、VSCode拡張機能マーケットプレイスからインストールする方法があります。

- アクティビティバーから拡張機能を選択します



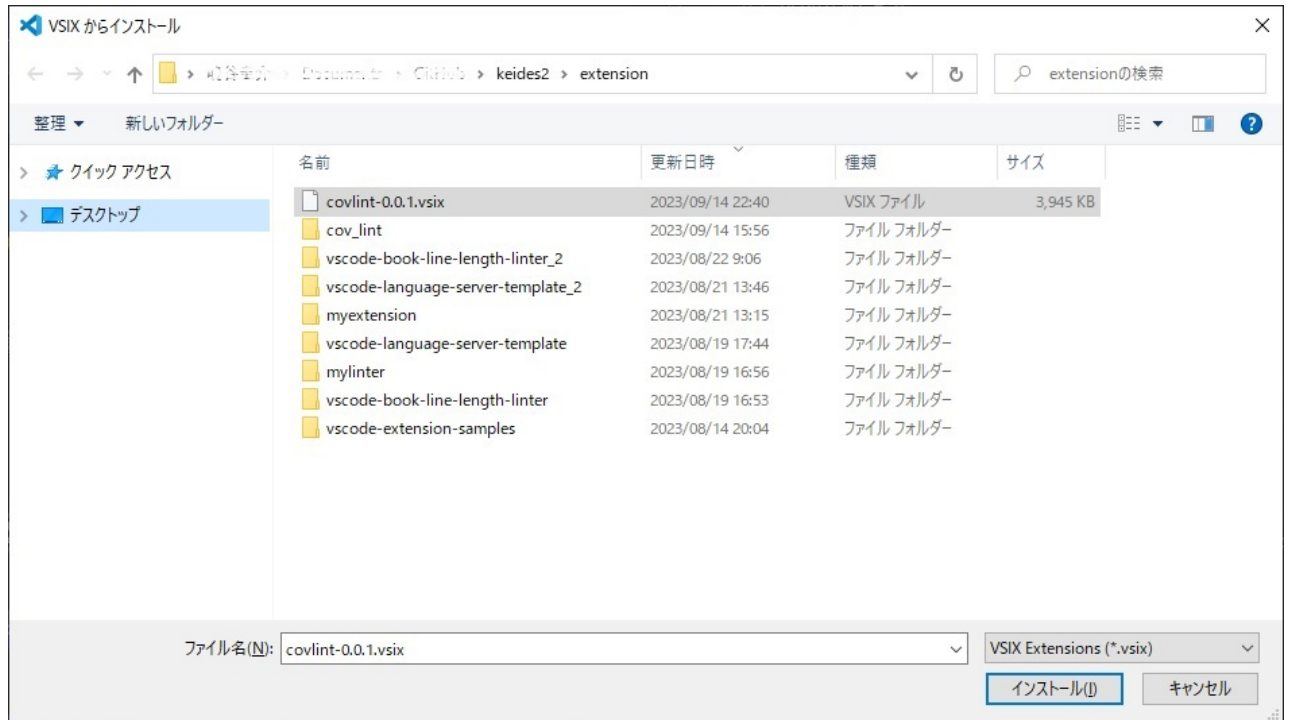
## 1.1 VSIXパッケージファイルからのインストール

- 3点メニューから「VSIXからインストール」を選択します

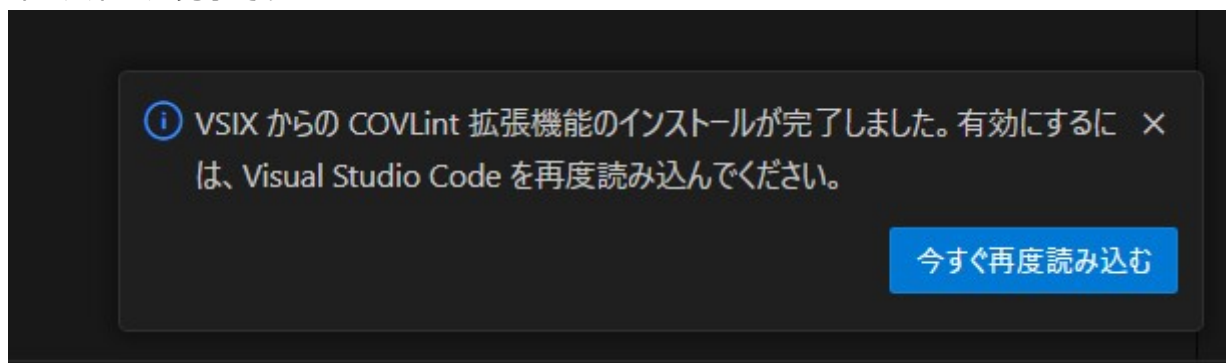




- ファイル「covlint-X.Y.Z.vsix」をインストールします



- インストール完了です

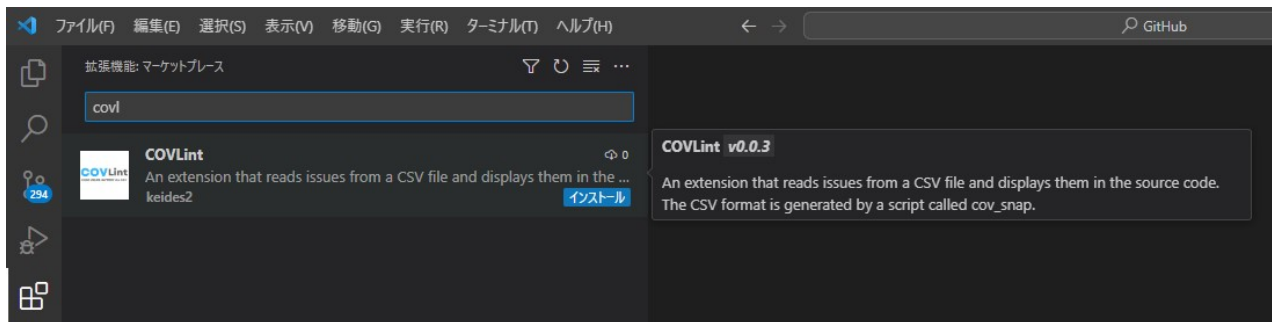


- 拡張機能 COVLint



## 1.2 VSCode拡張機能マーケットプレイスからのインストール

- 検索窓にcovlintを入力し、「インストール」を押下げます



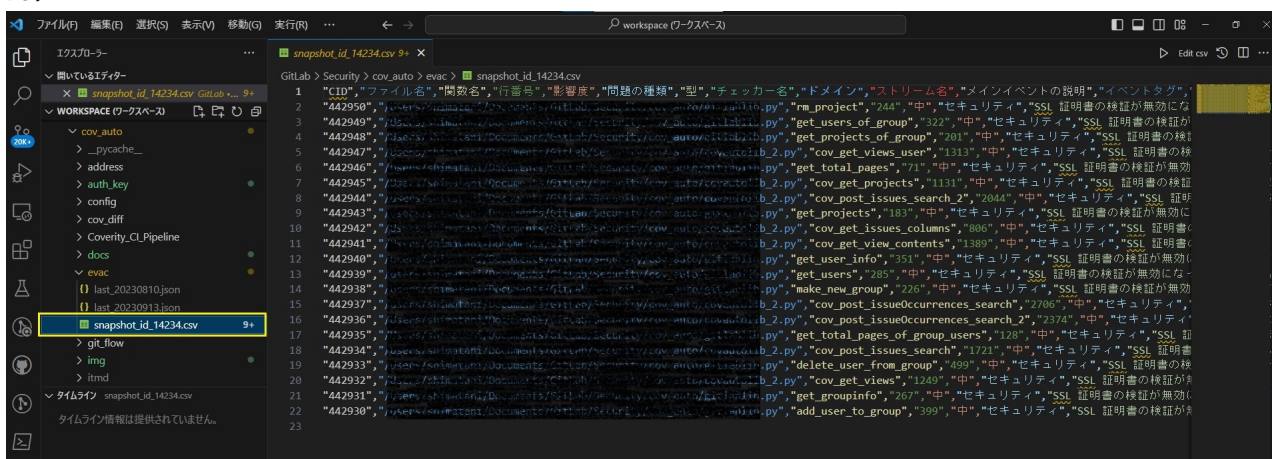
- インストール完了です



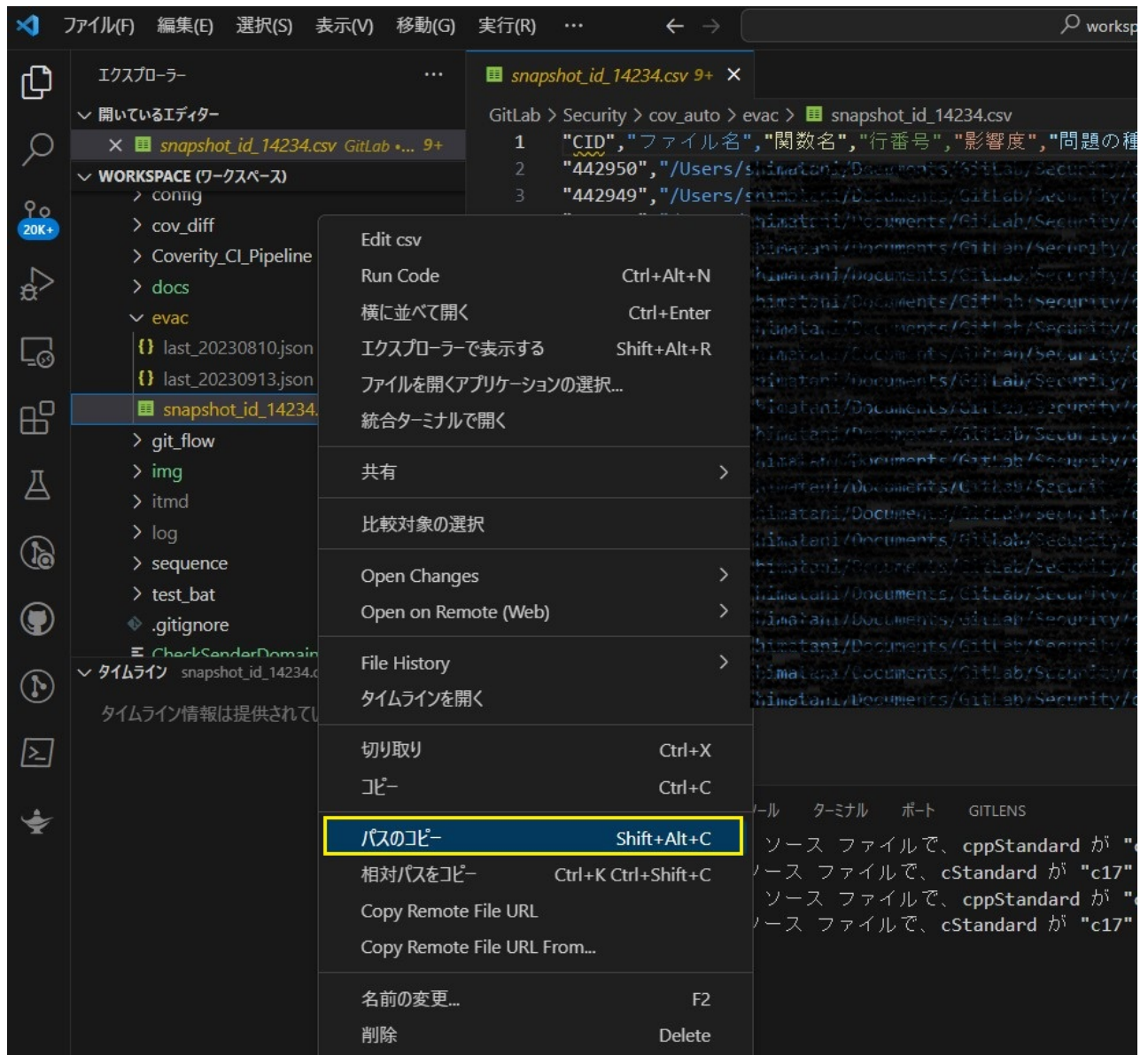
## 2. CSVファイルの読み込み

cov\_snapで取得したスナップショットCSVファイルを読み込みます

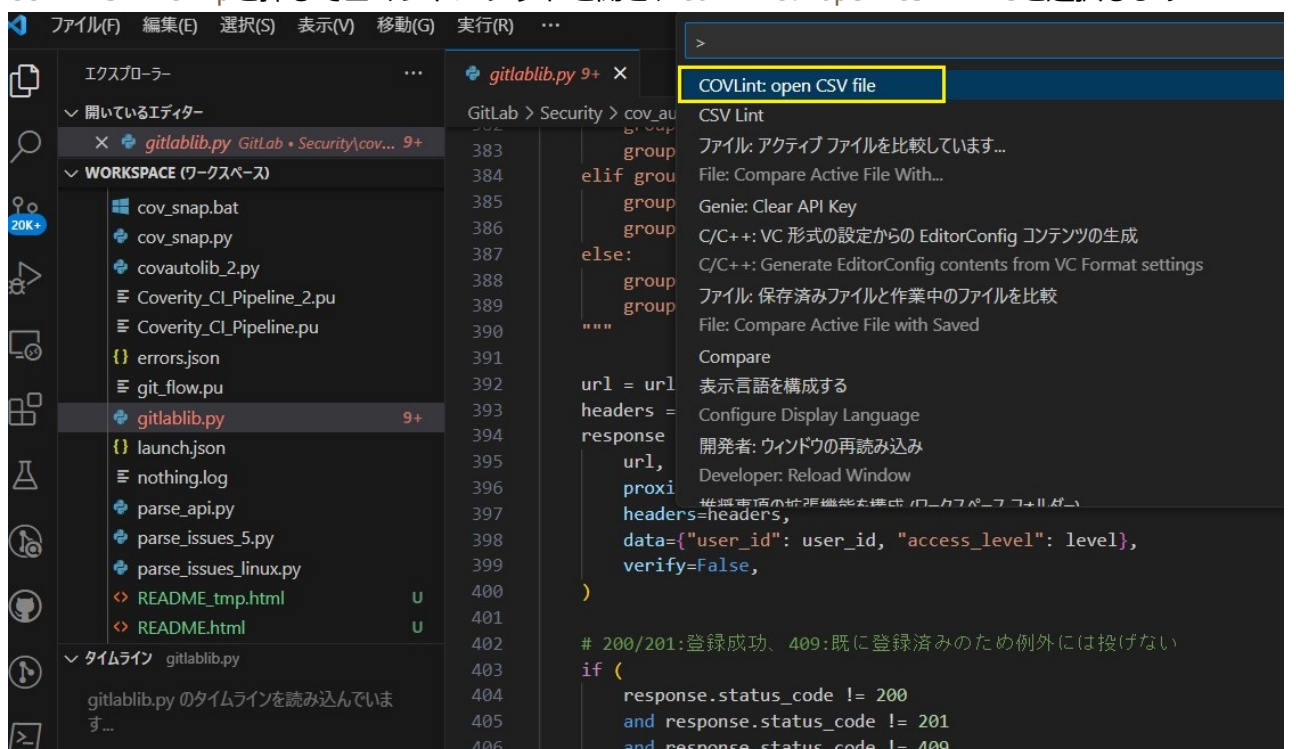
- VSCodeのエクスプローラーからスナップショットを保存しているフォルダに移動し、読み込みむスナップショットを**右クリック**で選択します（左クリックでファイルの中身を表示する必要はありません）



- メニューから「パスのコピー」を選択します

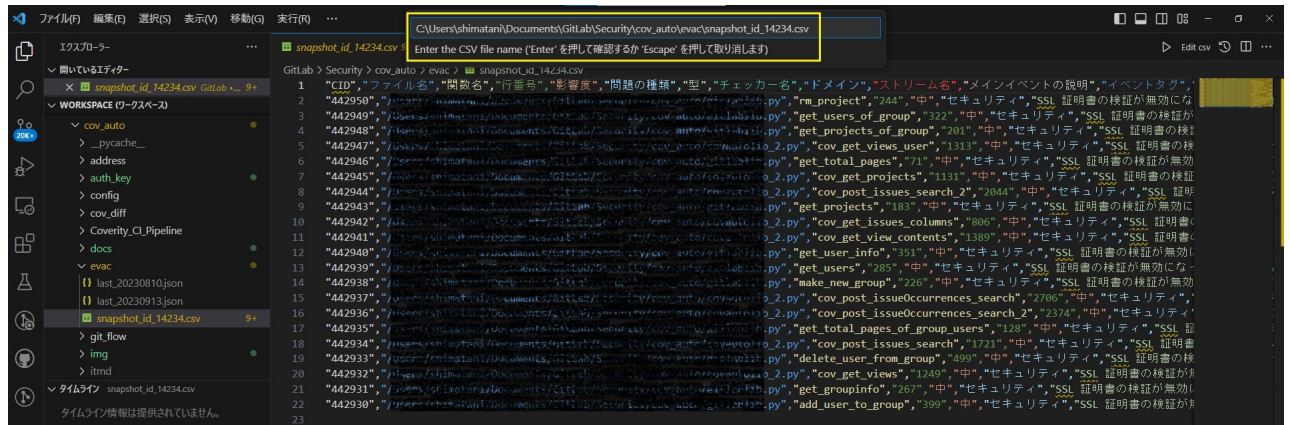


- Ctrl + Shift + pを押してコマンドパレットを開き、COVLint: open CSV fileを選択します

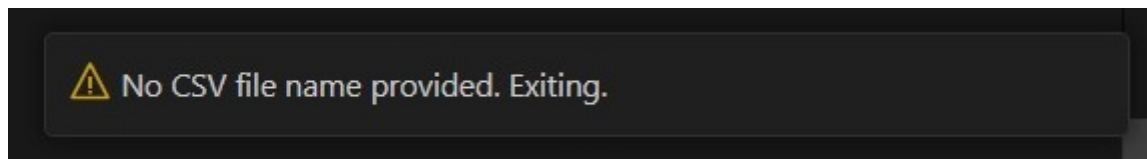




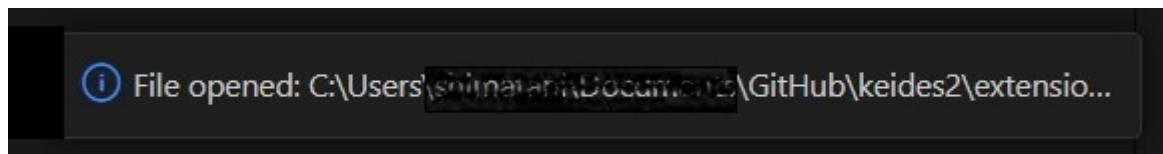
- 入力ボックスにコピーしたスナップショットCSVファイルのパスを貼り付けます



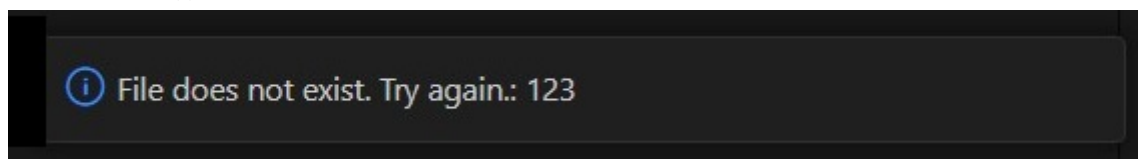
- 何も入力しなかったときのメッセージです



- 読み込みに成功したときのメッセージです



- 読み込みに失敗したときはリトライしてください



### 3. 指摘の表示

- フォルダを移動して開発中のソースコードを選択します

```

387     else:
388         group_name = "None"
389         group_id = 0
390         """
391
392     url = url_prefix + "groups/" + str(group_id) + "/members"
393     headers = {"Private-Token": private_token}
394     response = requests.post(
395         url,
396         proxies=proxies_dic,
397         headers=headers,
398         data={"user_id": user_id, "access_level": level},
399         verify=False,
400     )
401
402     # 200/201:登録成功、409:既に登録済みのため例外には投げる
403     if (
404         response.status_code != 200
405         and response.status_code != 201
406         and response.status_code != 409
407         and response.status_code
408         != 400 # グループから継承されたメンバーシップは Owner と同等またはそれ以上である必要があります
409     ):
410         raise AddGitLabException(
411             response.content,

```

- ソースコードの問題のある行に波線が入っているのでマウスオーバーします
  - 指摘がポップアップ表示されます
  - 問題パネルに指摘の一覧が表示されます
  - 問題を左クリックすると該当行にジャンプします

CID: 442930: SSL 証明書の検証が無効になっています (BAD\_CERT\_VERIFICATION)  
 ssl\_verify\_disabled: パラメータ verify は requests.post 関数で明示的に False に設定されています。  
 covlint

問題の表示 (Alt+F8)    クイックフィックス... (Ctrl+.)

問題 (122)    出力    ポート    デバッグコンソール    ターミナル    GITLENS    フィルター (例: テキスト、\*\*/\*.ts、!\*\*/\*node\_modules/\*\*)

- ⚠️ TIC is all uppercase. ex [Ln 386, Col 35]  
 gitlablib.py[Ln 386, Col 35]: Spelling matters  
 gitlablib.py[Ln 386, Col 35]: Particularly for names
- 💡 CID: 442930: SSL 証明書の検証が無効になっています (BAD\_CERT\_VERIFICATION) covlint [Ln 399, Col 9] ^  
 ssl\_verify\_disabled: パラメータ verify は requests.post 関数で明示的に False に設定されています。
- ⚠️ ID is all uppercase. ex [Ln 419, Col 13]  
 gitlablib.py[Ln 419, Col 13]: Spelling matters  
 gitlablib.py[Ln 419, Col 13]: Particularly for names
- ⚠️ KDEV is all uppercase. ex [Ln 421, Col 37]  
 gitlablib.py[Ln 421, Col 37]: Spelling matters

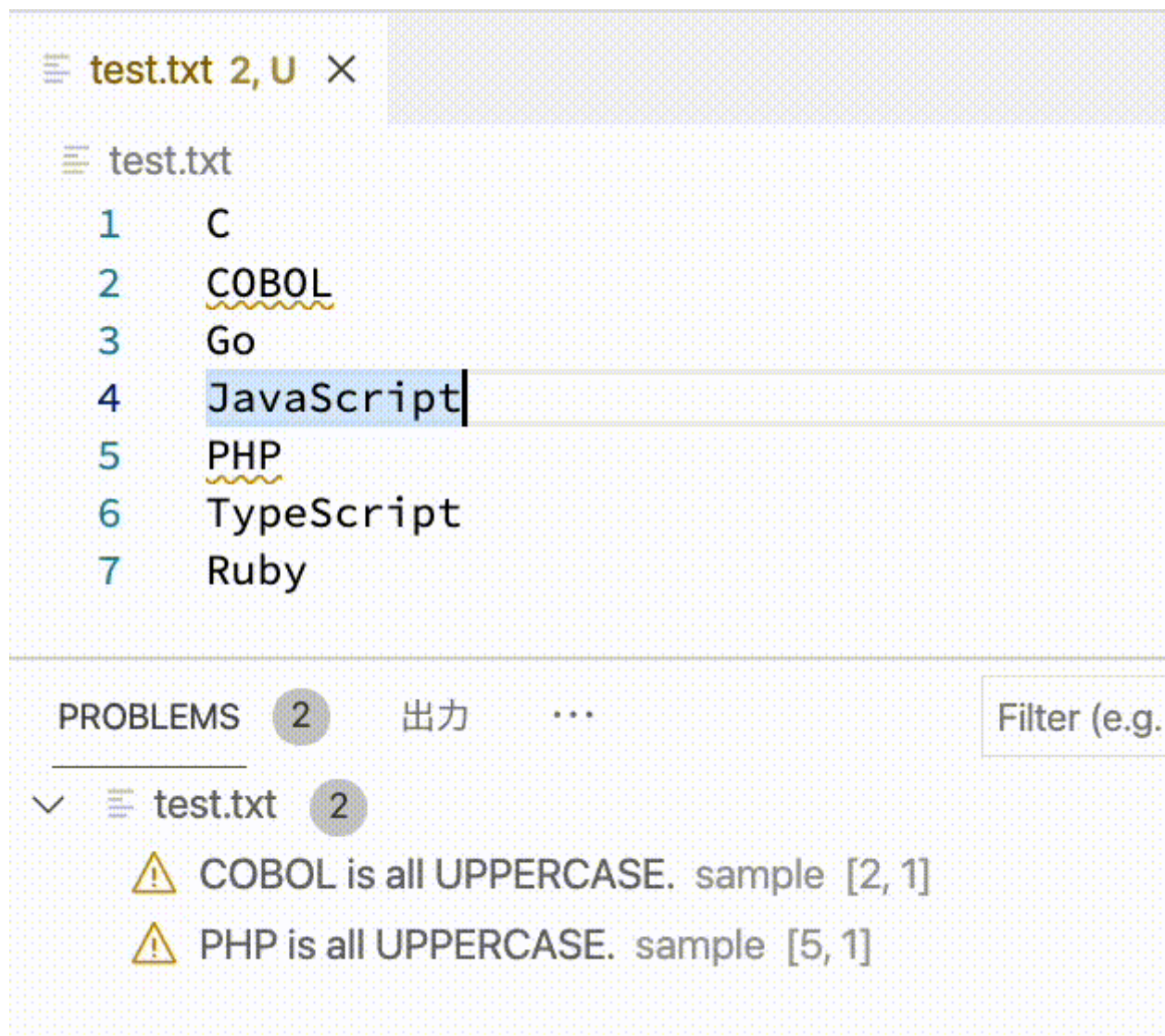


## 備考

Microsoftの<https://github.com/Microsoft/vscode-extension-samples>にあるlsp-sampleや、@Ikuyadeu 氏の記事 [Language Server Protocol開発チュートリアル](#) をベースにしましたので次の機能が残っています。

- コード検証機能
  - 3文字以上の大文字を検出します
- コード修正機能
  - 警告箇所を小文字に修正します

下図はvscode-language-server-template Linter README から。



- 
- 2023/09/15 keides2 v0.0.1 初版
  - 2023/09/16 keides2 v0.0.2 図の差し替え
  - 2023/09/17 keides2 v0.0.3 拡張機能マーケットプレイス対応