

# TDS-Dito: Ideias e coisas a fazer na extensão

---

## Legendas:



Pronto



Em andamento, sem prazo





Em estudo/a fazer (pode ser cancelado)

## Inteligência Artificial (Carol Clockin Web API)




- API para uso da inteligência artificial.
- Definir tratamento para **What do you want me to explain?** ou **Explain What?**, inclusive de códigos já processados.
- Apresentado lista de opções ao final da resposta (**Show me...**). Como proceder?
- Eliminar frases duplicadas.
- Eliminar sugestões de código já existente.
- Adicionar tempo de processamento em todas as funções da API.
- Configuração do nível de detalhes das respostas.  
Nessa etapa, apresentar sempre de forma sucinta, com opção para detalhamento.
- Controle de acesso (*login*).
- Configuração do idioma das respostas.
- Adicionar suporte para outros idiomas.
- Em alguns retornos de explicação de um bloco, retornou referências a linha (**First line, Second Line...**). Melhor seria textos mais corridos, sem ser linha a linha e de acordo com nível de detalhe desejado.
- Mecanismo de avaliação de respostas pelo usuário, que poderá ser utilizada para refinamento do modelo.
- API de log e avaliação de auto-complete

## API: Inteligência Artificial (interface IaApiInterface)






- API para a inteligência artificial.
- Identificação de usuário (*login*).  
Usar **vscode.AuthProvider** para autenticação.

-  Mensagens associadas a processamento.  
Encapsular o processamento e passar como *callback* para ChatApi. Dito e esse passa a tratar o retorno da mensagem (*messageId*).
-  No caso de erro 504 e com informação de tempo para nova tentativa, agendar nova tentativa.  
Mensagem com tempo: *The server encountered a temporary error and could not complete your request. Please try again in 30 seconds.*





## Extensão

-  Implementar sistema de tradução L10N.
-  Implementar uma forma de *feedback* do resultado do auto-complete.
  - Caso o usuário aceite o auto-complete, gerar um log com avaliação positiva.
  - Pensar em alguma solução para o usuário, ativamente, avaliar negativamente o auto-complete sugerido pela IA.
-  Definir processo de identificação de usuário (*login*).

## API: Bate-papo (classe ChatApi)

-  API para bate-papo.
-  Detalhar a *ajuda(help)* dos comandos.
-  Melhorar/implementar tratamento de argumentos em comandos.
-  Implementar processo de abertura de chamado (comando *open\_issue*).
-  Implementar substituição de texto ao fazer o typify



## Visual: Bate-papo

-  Executar o comando ao acionar *Enter*.
-  Associar visualmente mensagem de resposta com a mensagem de entrada.  
Tratar *MouseOver* sobre a mensagem de resposta ou de entrada e destacar as duas. Dessa forma, se o usuário disparar vários processos, pode acompanhar qual mensagem está sendo respondida.
-  Implementar *goto* em ligações (*links*) de posicionamento em fontes.
-  Implementar indicador de "Processando" durante execução da API *complete*.  
Ideias: *Completing code...* ou barra de progresso (*vscode.window.withProgress*) ou outro indicador acima do campo *NewMessage*.

## Visual: Editor de texto

-  Colocar *codeLens* no código que esta sendo processado.

## Problemas gerais

-  Extensão não ativa corretamente caso a visão do DitoChat tenha sido fechada anteriormente
-  Abrir DitoChat no painel secundário