

**Руководство пользователя
Ассистент для команд разработки**

СОДЕРЖАНИЕ

1. ВВЕДЕНИЕ	3
2. ТЕРМИНЫ И СОКРАЩЕНИЯ	4
3. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ИИ-ЧАТОМ В БРАУЗЕРЕ	5
3.1. Начало работы	5
3.3. Создание, редактирование и удаление подсказок для модели (промpts)	7
3.4. Ввод запроса	7
3.5. Оценка ответа	8
3.6. Очищение истории	9
3.7. Генерация API-ключей для аутентификации пользователей	9
3.8. Добавление файлов в контекст запроса	9
4. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПЛАГИНОМ ДЛЯ IDE VS CODE	10
4.1. Установка плагина для IDE VS Code	10
4.2. Авторизация	11
4.3. Создание нового чата	12
4.4. Удаление чатов	12
4.5. Генерация кода и оптимизация	13
4.6. Генерация unit-тестов	13
4.7. Генерация комментариев	14
4.8. Автодополнение кода	15
5. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПЛАГИНОМ ДЛЯ IDE INTELLIJ IDEA	15
5.1. Установка плагина для IDE IntelliJ IDEA	15
5.2. Авторизация	16
5.3. Создание нового чата	17
5.4. Удаление чатов	17
5.5. Генерация кода и оптимизация	17
5.6. Генерация unit-тестов	18
5.7. Генерация комментариев	19
5.8. Автодополнение кода	20

1. ВВЕДЕНИЕ

Система «Ассистент для команд разработки» (далее - Система) автоматизирует генерацию программного кода, кода тестов ПО, написание сценариев тестирования и текста для технической и сопровождающей документации на ПО, используя ИИ для анализа введенной Пользователем информации (запросы на естественном языке, требования, блоки кода) и генерации качественных ответов от LLM (оптимизированного кода, тестов и текста на основе шаблонов). Система интегрируется в вашу среду разработки или может быть открыта в веб-браузере, что повышает скорость разработки программного кода, снижает количество ошибок при его написании и улучшает качество кода.

Основные функции и возможности:

1. Для аутентификации пользователей Система позволяет генерировать API-ключи.
2. Система поддерживает текстовый чат с возможностью отправки Пользователем запросов к LLM и получения ответов от LLM, поддержкой контекста и истории диалога.
3. Система обеспечивает возможность изменения и сохранения параметров генерации ответов LLM в текстовом чате (температура, длина ответа, глубина контекста).
4. Система позволяет генерировать программный код на основе текстового описания задачи. Использование данной функции возможно в веб-интерфейсе или интегрированной среде разработки (IDE):
 - Генерация кода по описанию задачи на естественном языке;
 - Оптимизация и рефакторинг существующего кода;
 - Генерация комментариев к коду.
5. Система автоматически генерирует unit-тесты для конкретных частей программного кода, что позволяет сократить время тестирования:
 - Генерация unit-тестов.
6. Система автоматизирует такие задачи, как документирование кода и генерация комментариев:
 - Генерация комментариев к коду;
 - Автоматическое код-ревью.
7. Система обеспечивает возможность загружать документацию в качестве дополнительных знаний для LLM в различных форматах (.txt, .java, .js, .ts, .html, .MD).

2. ТЕРМИНЫ И СОКРАЩЕНИЯ

В тексте настоящего документа представлены следующие сокращения (см. **Таблица 1-1**).

Таблица 1-1 — Перечень сокращений

Сокращение/аббревиатура	Значение
ИИ	Искусственный интеллект
АКР	Ассистент для команд разработки
Система	Ассистент для команд разработки
IDE	Интегрированная среда разработки
LLM	Большая языковая модель

В тексте настоящего документа представлены следующие термины и определения (см. **Таблица 1-2**).

Таблица 1-2 — Перечень терминов и определений

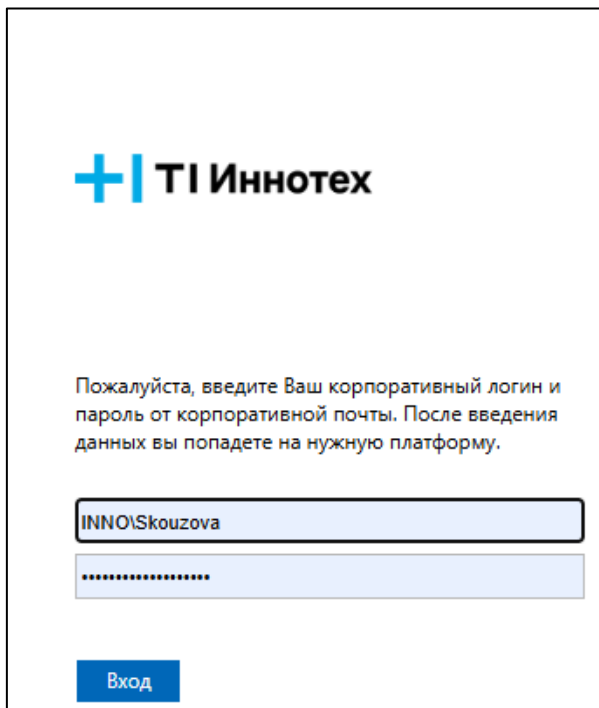
Аутентификация	Проверка подлинности предъявленного Пользователем идентификатора
Администратор Системы	Пользователь Системы с ролью «Администратор»
Заказчик	Физическое или юридическое лицо, которое инициирует, заказывает и финансирует разработку, модификацию или поддержку программного обеспечения Системы. После приемки услуг по разработке Системы Заказчик является Владелец Системы
Пользователь	Работник, наделенный правами доступа к информационным ресурсам организации
Подсказка	Системный промпт для LLM - представляет собой руководящие инструкции или исходные данные, которые задаются модели, чтобы она могла генерировать ответы на запросы пользователя
Роль	Набор полномочий, который необходим Пользователю для выполнения определённых рабочих задач. Каждый сотрудник может иметь одну или несколько ролей, а каждая роль может содержать от одного до множества полномочий, которые разрешены Пользователю в рамках этой роли. Роли Пользователей в Системе: – «Пользователь»; – «Администратор».
ИТ-инфраструктура	Совокупность всего программного обеспечения, оборудования, сетей и подключенных сервисов, образующих ИТ-среду организации
Файл	Цифровой носитель информации, содержащий текстовые данные, с любым из перечисленных расширений: .txt, .java, .js, .ts, .html, .MD.

3. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ИИ-ЧАТОМ В БРАУЗЕРЕ

3.1. Начало работы

Для начала работы с ИИ-чатом необходимо ввести в браузере IP-адрес или URL, полученный от администратора системы, после чего откроется страница начала работы с Системой.

Далее необходимо выполнить аутентификацию при помощи корпоративной системы аутентификации.

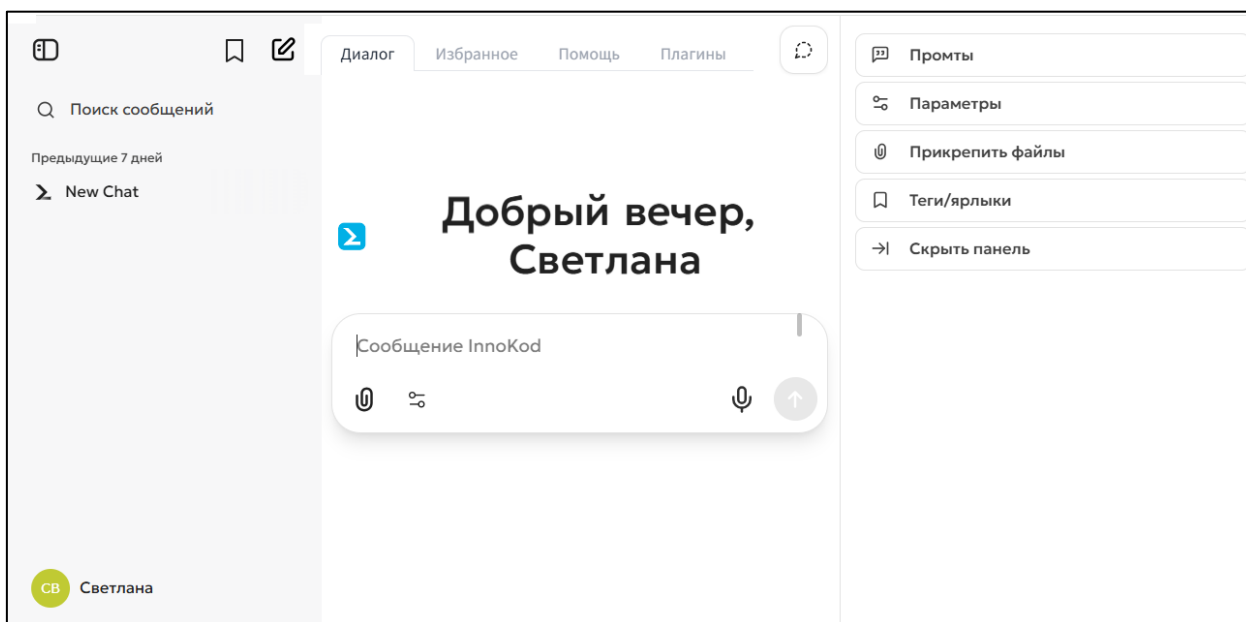


После ввода логина и пароля имеющейся учетной записи выполняется переход на начальную страницу Системы.

Для получения доступа к Системе необходимо обратиться к Администратору.

Начальная страница Системы состоит из следующих элементов:

- Верхняя навигационная панель управления, на которой расположены основное меню системы с доступными Пользователю вкладками;
- Основная рабочая область – в этой области отображается диалог с ИИ-чатом и поле для ввода запроса;
- Нижняя навигационная панель управления, на которой расположены элементы управления для взаимодействия с ИИ-чатом;
- Меню, в котором можно настроить параметры и системный промпт.



3.2. Настройка параметров модели

Настройка параметров модели позволяет адаптировать Систему под конкретные задачи и требования проекта. Система предоставляет гибкие возможности для настройки параметров генеративной модели, что позволяет улучшить качество и релевантность генерируемых решений.

Для настройки параметров модели необходимо перейти в раздел «Параметры».

Параметры

Генерация

Температура (креативность)

0,22

top-p (nucleus sampling), эрудиция

0,99

Freq Penalty, штраф высокочастотных токенов

0,02

Presence Penalty, штраф за повторения

0

Максимальная длина одного вопроса+ответа (в токенах)

1000

Учитывать N прошлых сообщений в диалоге (0 - отвечать без истории)

2

Интерфейс

☒ Загружать прошлый чат при загрузке

☒ Показывать панель подсказок с примерами при загрузке

Сбросить Параметры модели

Сохранить как Пресет

Пользователь может задавать указанные параметры в соответствии с собственными требованиями к ответу модели.

По умолчанию для данных параметров заданы оптимальные (с точки зрения скорости и качества ответов LLM) значения.

3.3. Создание, редактирование и удаление подсказок для модели (промpts)

Системный промпт — это начальные установки, которые задают контекст для взаимодействия с LLM. Пользователь может настроить промпт в соответствии с требованиями своего проекта, открыв пункт меню «Системный промпт». В открывшейся области описания системного промпта необходимо ввести собственный текст, либо отредактировать предустановленный.

Системный промпт

По умолчанию в Системе указан оптимальный промпт для разработчиков программного кода.

3.4. Ввод запроса

Чат-бот позволяет быстро получать ответы на вопросы по коду, документации и техническим требованиям.

В текстовое поле необходимо ввести вопрос или задачу, которую нужно решить. Чем точнее и детальнее будет запрос, тем более релевантный ответ будет получен. После ввода вопроса необходимо нажать кнопку «Отправить» (стрелка вверх).

СВ

Светлана
Напиши функцию на Java, которая принимает два целочисленных параметра и возвращает их сумму

IK

InnoKod
Конечно! Вот пример функции на Java, которая принимает два целочисленных параметра и возвращает их сумму:

java

Копировать код

```
public class SumCalculator {  
    // Метод для вычисления суммы двух целых чисел  
    public static int sum(int a, int b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        // Пример использования метода  
        int result = sum(5, 7);  
        System.out.println("Сумма: " + result); // Вывод: Сумма: 12  
    }  
}
```

В этом примере:

- Класс `SumCalculator` содержит метод `sum`, который принимает два целых числа `a` и `b` и возвращает их сумму.
- В методе `main` показано, как вызвать функцию `sum` и вывести результат.

🔊

📄

✎

🔗

👍

👎

🔄

Чтобы получить максимально полезный ответ от чат-бота, необходимо следовать рекомендациям:

- 1) Формулировать конкретный вопрос: указывать язык программирования, фреймворк или технологию, с которой выполняется работа. Если запрос связан с кодом, указать контекст или приложить фрагмент кода;
- 2) Использовать естественный язык: формулировать запросы на естественном языке, как объяснение задачи коллеге;
- 3) Указывать дополнительные параметры: если существуют конкретные требования к стилю кода, производительности или безопасности, необходимо указать их в запросе.

Пользователю предоставляется возможность редактировать запрос. Для этого необходимо нажать на иконку «Карандаш» под запросом внутри диалога.

СВ

Светлана
Напиши функцию на Java, которая принимает два целочисленных параметра и возвращает их сумму

🔊

📄

✎

🔗

3.5. Оценка ответа

Чтобы оценить ответ ИИ-чата, необходимо нажать на соответствующую иконку: 👍 («Нравится») или 👎 («Не нравится»). Это поможет улучшить качество предоставляемых ответов и адаптировать систему под предпочтения Пользователя.

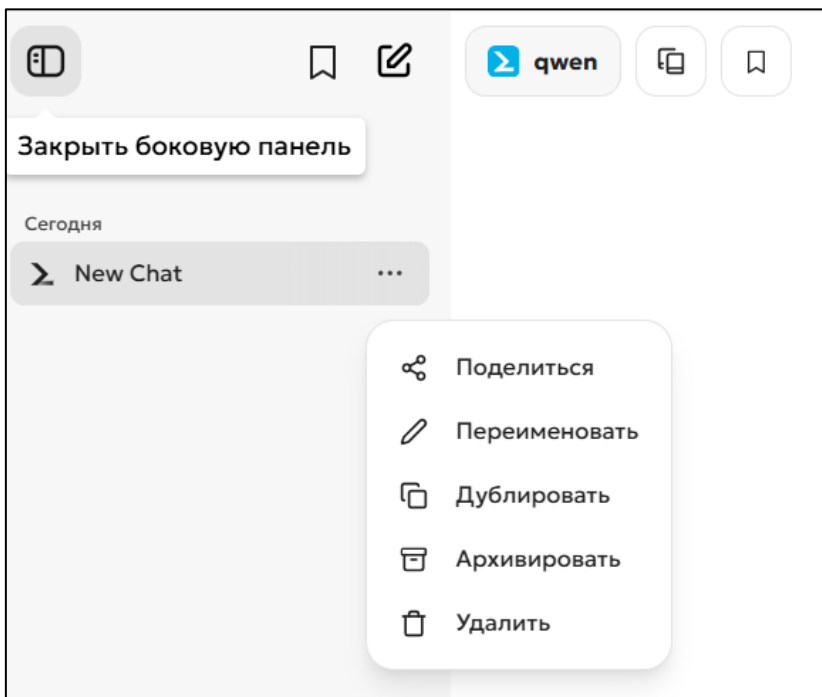
В этом примере:

- Класс `SumCalculator` содержит метод `sum`, который принимает два целых числа `a` и `b` и возвращает их сумму.
- В методе `main` показано, как вызвать функцию `sum` и вывести результат.

🔊 📄 ✎ 🧩 👍 🗑️ ↺

3.6. Очистение истории

Чтобы удалить историю ИИ-чата, необходимо открыть контекстное меню конкретного диалога и выбрать «Удалить».



3.7. Генерация API-ключей для аутентификации пользователей

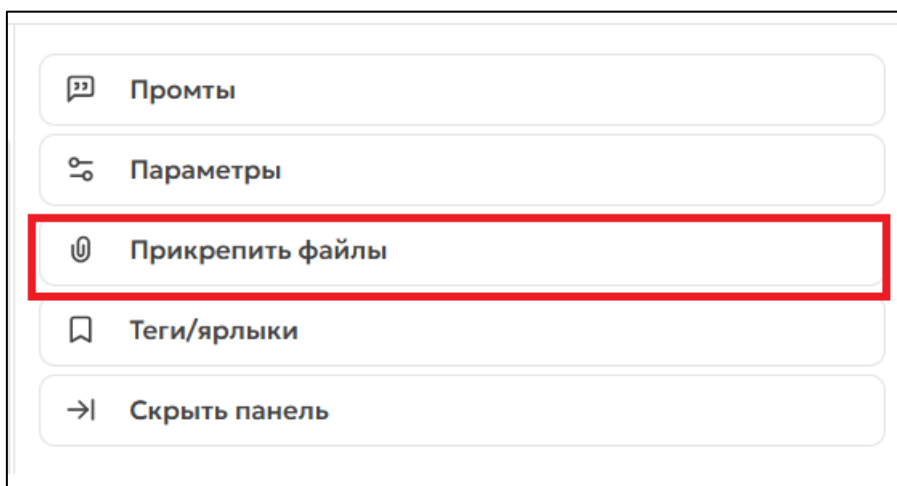
Для получения API-ключа необходимо перейти на вкладку «Плагины для IDE» и нажать на кнопку «Показать ваш ключ API»

Ваш персональный ключ (не передавайте никому):

Показать ваш ключ API

3.8. Добавление файлов в контекст запроса

Для загрузки файлов и добавления их содержания в контекст запросов необходимо открыть пункт меню «Прикрепить файлы» и добавить файл любого формата из следующего списка: .txt, .java, .js, .ts, .html, .MD.



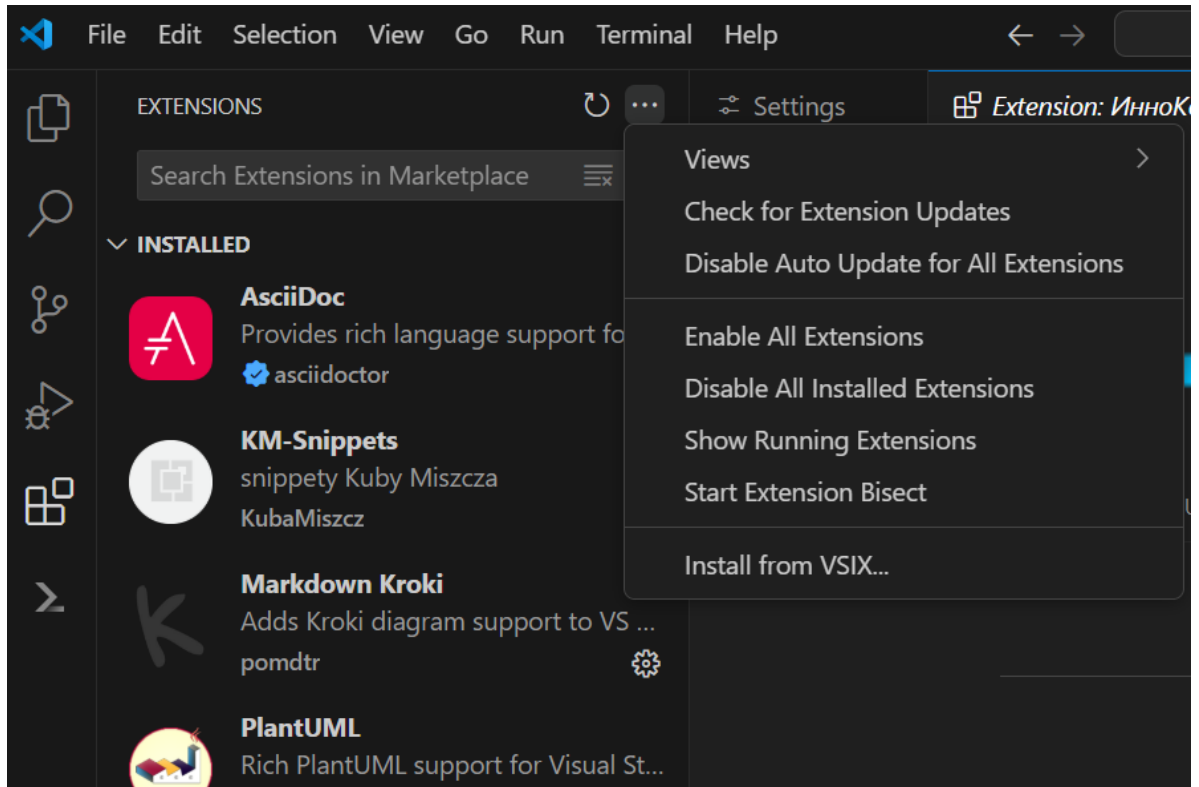
4. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПЛАГИНОМ ДЛЯ IDE VS CODE

Плагин Системы для IDE VS Code позволяет взаимодействовать с функциональностью Системы через интегрированную среду разработки.

4.1. Установка плагина для IDE VS Code

Шаги для установки:

- Необходимо открыть ИИ-чат Системы в веб-браузере (порядок действий описан в п.3 настоящего документа);
- Открыть вкладку «Плагины»;
- Нажать на кнопку «Скачать плагин VSCode», выбрав версию для своей операционной системы. После нажатия кнопки «Скачать» файл с расширением .VSIX будет автоматически загружен на локальное устройство Пользователя;
- Открыть окно IDE VS Code, выбрать вкладку «Extentions», в открывшемся списке расширений IDE открыть контекстное меню (три точки справа сверху);



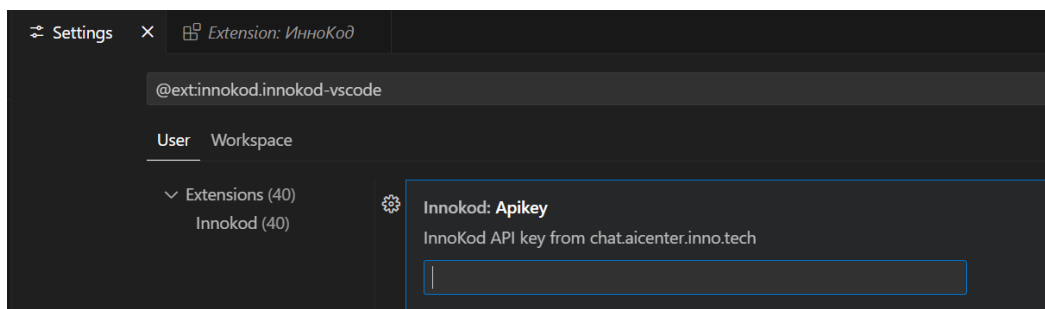
- В открывшемся окне выбрать пункт «Install from VSIX...», затем выбрать файл .VSIX, который был скачан на предыдущих шагах, и нажать кнопку «Install»;
- После установки плагина в интерфейсе IDE VS Code слева появится иконка со стрелкой, направленной вправо – плагин Системы.

4.2. Авторизация

После установки плагина необходимо авторизоваться в Системе.

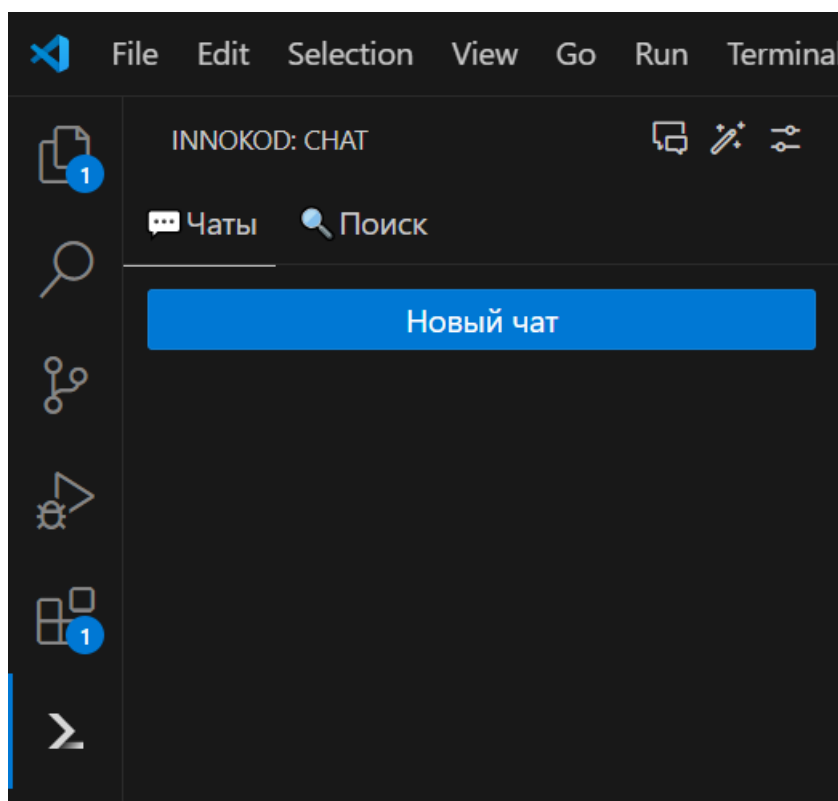
Для этого:

- Необходимо открыть ИИ-чат Системы в веб-браузере (порядок действий описан в п.3 настоящего документа);
- Открыть вкладку «Плагины», нажать на кнопку «Показать ваш ключ API», скопировать значение ключа;
- Открыть окно IDE VS Code и нажать на иконку Системы;
- В открывшейся области «Explorer» из контекстного меню справа сверху выбрать «Настройки»;
- В строку «API Key» выполнить вставку скопированного на предыдущих шагах значения API Key и сохранить изменения настроек. После сохранения изменений настроек функциональность Системы в плагине для IDE VS Code станет доступной для Пользователя.

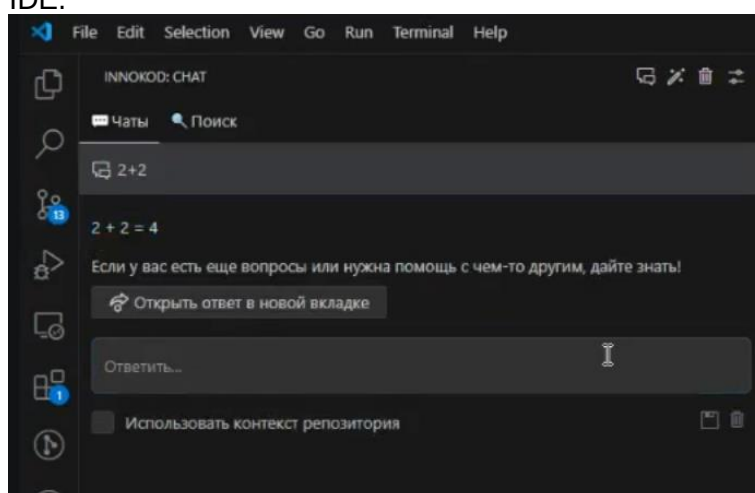


4.3. Создание нового чата

- В области «Explorer» IDE VS Code необходимо нажать на кнопку «Новый чат». После нажатия на кнопку будет открыта область для начала диалога;

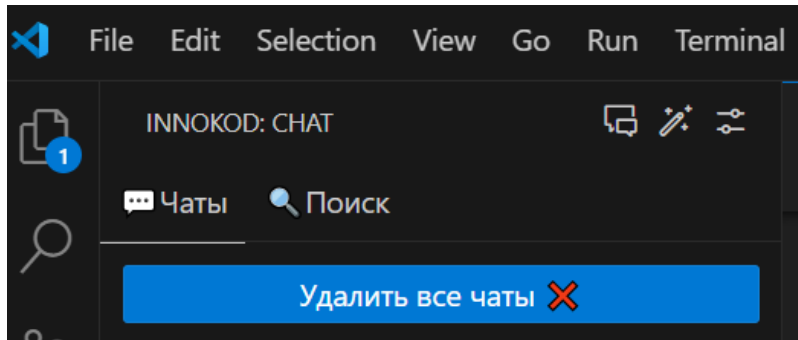


- Ввести вопрос на русском или английском языке в области «Спросить» и нажать на кнопку Enter;
- После ввода вопроса чат выполняет генерацию и отображение ответа в интерфейсе IDE.



4.4. Удаление чатов

- В области «Explorer» IDE VS Code необходимо нажать на кнопку «Удалить все чаты»;



- После нажатия на кнопку «Удалить все чаты» - история диалогов будет очищена.

4.5. Генерация кода и оптимизация

Плагин для IDE позволяет автоматически генерировать код на основе текстового описания задачи.

Порядок использования:

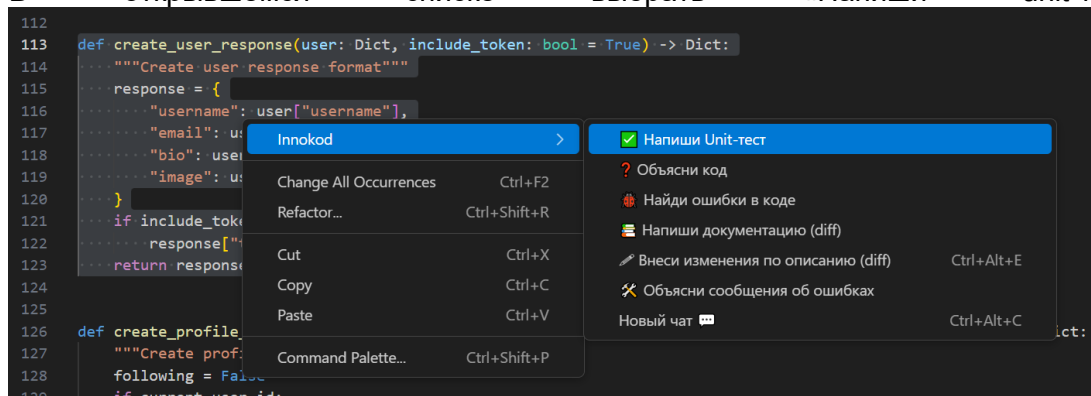
- На вкладке «Чаты» в области «Спросить» необходимо сформулировать запрос на генерацию кода или оптимизацию (например, «напиши функцию на Java, которая принимает два целочисленных параметра и возвращает значение их суммы» или вставить текст программного кода и написать «Предложи варианты улучшения этого кода») и нажать на кнопку Enter;
- В ответе чата будет предложен готовый текст программного кода или предложена улучшенная версия кода. Код из ответа чата может быть скопирован и вставлен в файл проекта.

4.6. Генерация unit-тестов

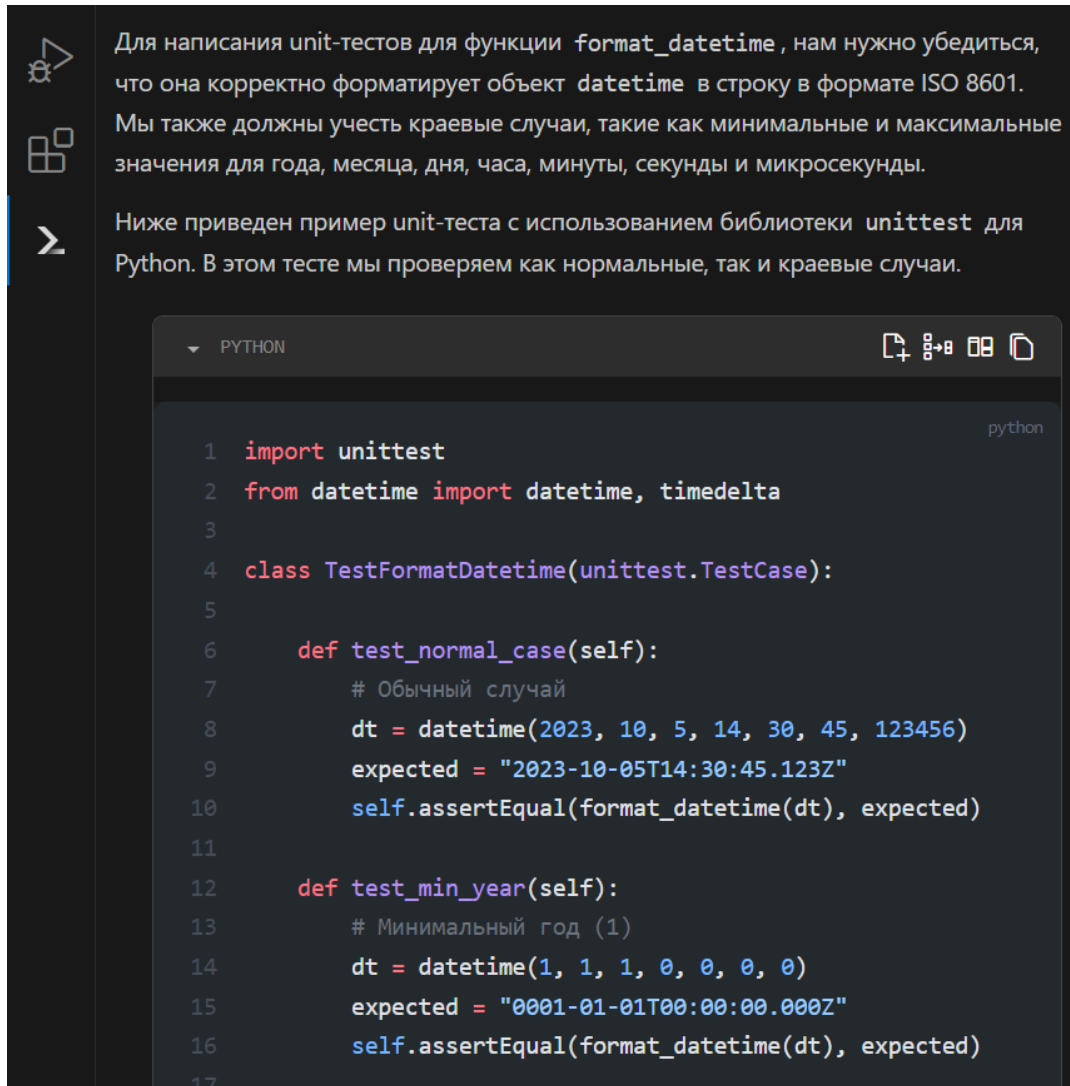
Плагин для интегрированной среды разработки позволяет сгенерировать unit-тесты для выбранной Пользователем части программного кода.

Порядок использования:

- В IDE VS Code открыть файл проекта, который содержит программный код, для которого необходимо сгенерировать unit-тест;
- В открытом файле выделить часть кода, для которой необходимо сгенерировать unit-тест, нажать правую кнопку мыши;
- В открывшемся списке меню выбрать «InnoKod»;
- В открывшемся списке выбрать «Напиши unit-тест»;



- После выбора данного пункта меню unit-тест будет сгенерирован в виде ответа в чате.

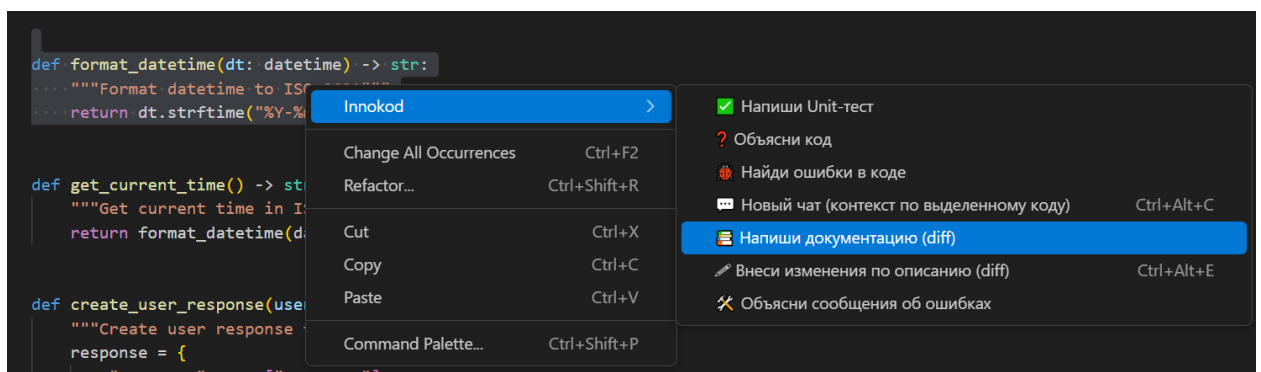


4.7. Генерация комментариев

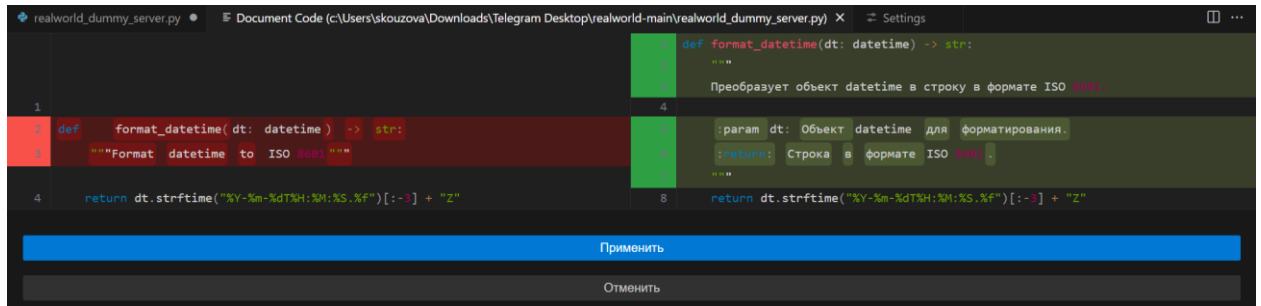
Плагин позволяет автоматически генерировать комментарии к выделенному программному коду.

Для запуска данной функции необходимо:

- Открыть файл с программным кодом в IDE VS Code;
- Выделить часть кода, к которой необходимо сгенерировать комментарий, нажать правой кнопкой мыши, выбрать пункт меню «InnoKod», затем «Напиши документацию»



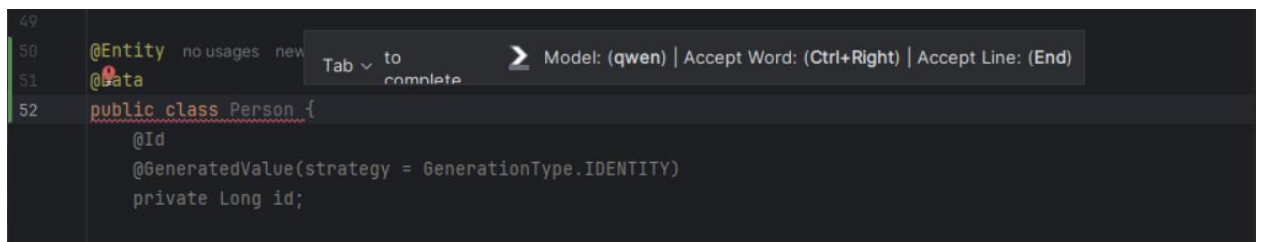
- После запуска данной функции автоматически будут предложены комментарии, которые Пользователь может применить или отменить.



4.8. Автодополнение кода

Плагин для интегрированной среды разработки выполняет автоматическое дополнение программного кода при написании кода в файле, отрытом в IDE.

Пример работы:



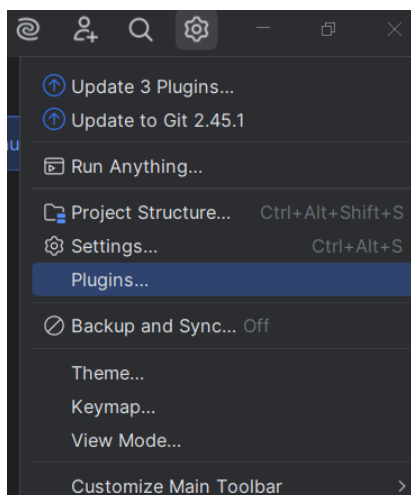
5. ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ С ПЛАГИНОМ ДЛЯ IDE INTELLIJ IDEA

Плагин Системы для IDE IntelliJ IDEA позволяет взаимодействовать с функциональностью Системы через интегрированную среду разработки.

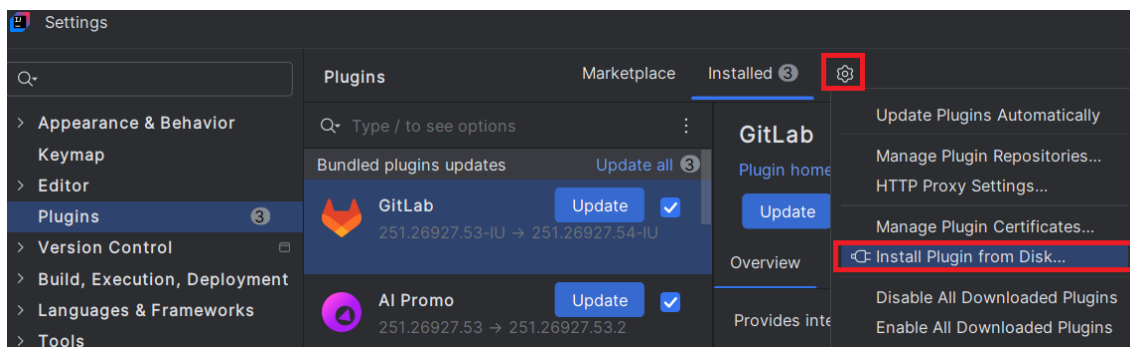
5.1. Установка плагина для IDE IntelliJ IDEA

Шаги для установки:

- Необходимо открыть ИИ-чат Системы в веб-браузере (порядок действий описан в п.3 настоящего документа);
- Открыть вкладку «Плагины»;
- Нажать на кнопку «Скачать плагин JetBrains». После нажатия кнопки «Скачать» файл с расширением .zip будет автоматически загружен на локальное устройство Пользователя;
- Открыть окно IDE IntelliJ IDEA, в контекстном меню справа сверху открыть Settings->Plugins;



- В контекстном меню открывшегося окна выбрать установку плагина с локального устройства Пользователя (Install Plugin from Disk);



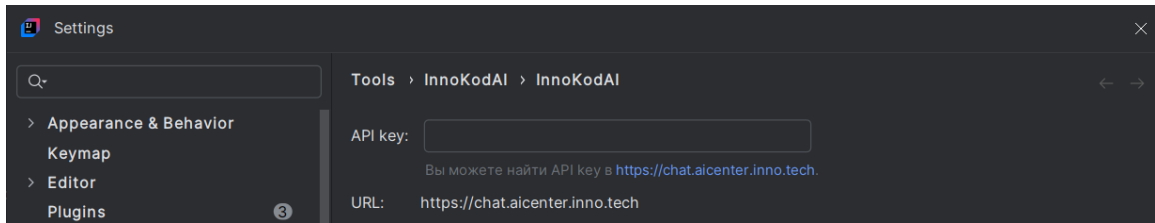
- Выбрать файл .zip, который был скачан на предыдущих шагах, и нажать кнопку «Открыть»;
- После установки плагина в интерфейсе IDE IntelliJ IDEA появится иконка со стрелкой, направленной вправо ➡ – плагин Системы.

5.2. Авторизация

После установки плагина необходимо авторизоваться в Системе.

Для этого:

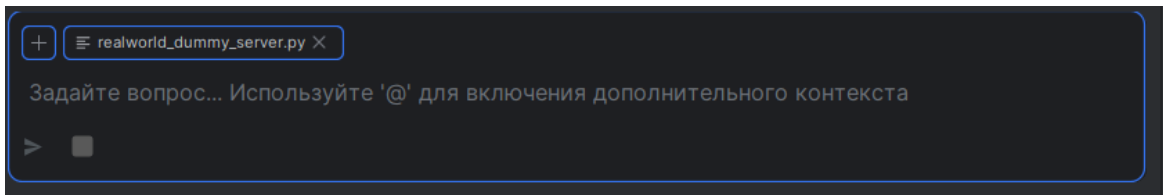
- Необходимо открыть ИИ-чат Системы в веб-браузере (порядок действий описан в п.3 настоящего документа);
- Открыть вкладку «Плагины», нажать на кнопку «Показать ваш ключ API», скопировать значение ключа;
- Открыть окно IDE IntelliJ IDEA и нажать на иконку плагина Системы;
- Выбрать «Открыть настройки»;
- В строку «API key» выполнить вставку скопированного на предыдущих шагах значения API Key и нажать «ОК». После сохранения изменений настроек функциональность Системы в плагине для IDE IntelliJ IDEA станет доступной для Пользователя.



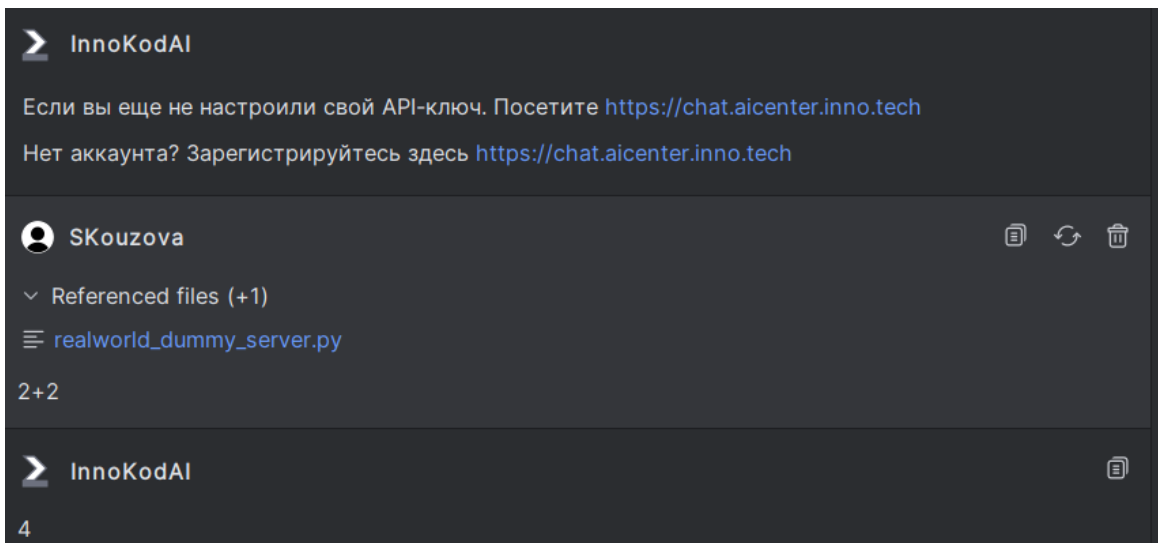
5.3. Создание нового чата

Для создания нового чата необходимо выполнить следующие шаги:

- Нажать на иконку плагина ;
- Найти область «Задайте вопрос...»;

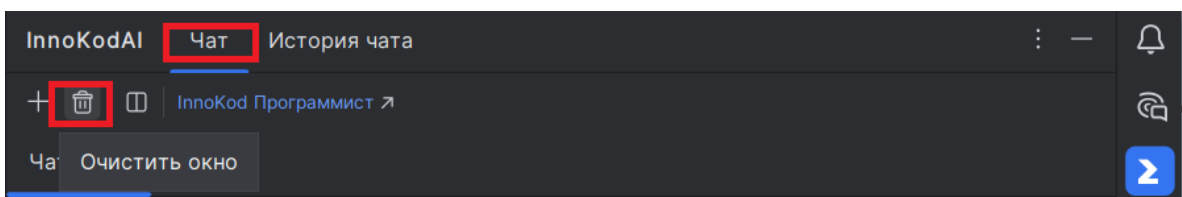


- Ввести в область «Задайте вопрос...» свой вопрос на русском или английском языке и нажать на иконку  или кнопку Enter на клавиатуре;
- После ввода вопроса чат выполняет генерацию и отображение ответа в интерфейсе IDE.



5.4. Удаление чатов

- Для удаления истории чата на вкладке «Чат» области работы с плагином необходимо нажать на иконку «Корзина».

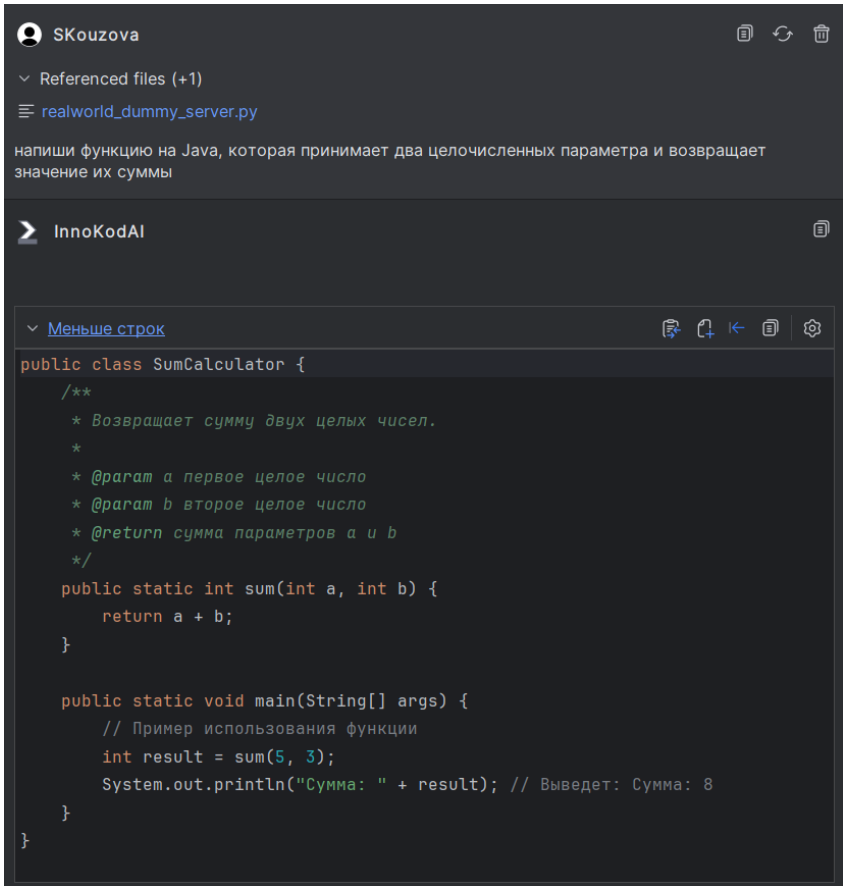


5.5. Генерация кода и оптимизация

Плагин для IDE позволяет автоматически генерировать код на основе текстового описания задачи.

Порядок использования:

- На вкладке «Чат» в области «Задайте вопрос...» плагина необходимо сформулировать запрос на генерацию кода или оптимизацию (например, «напиши функцию на Java, которая принимает два целочисленных параметра и возвращает значение их суммы» или вставить текст программного кода и написать «Предложи варианты улучшения этого кода») и нажать на кнопку Enter;
- В ответе чата будет предложен готовый текст программного кода или предложена улучшенная версия кода. Код из ответа чата может быть скопирован и вставлен в файл проекта.



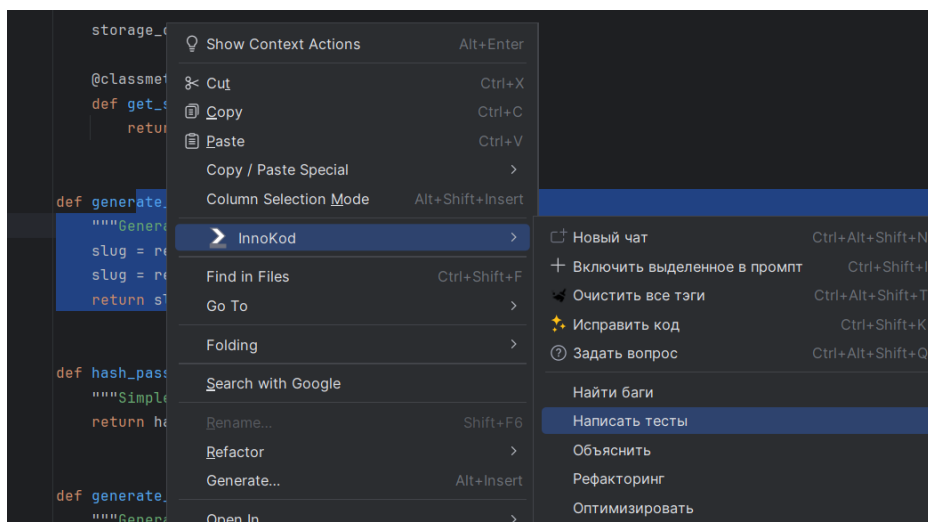
```
public class SumCalculator {  
    /**  
     * Возвращает сумму двух целых чисел.  
     *  
     * @param a первое целое число  
     * @param b второе целое число  
     * @return сумма параметров a и b  
     */  
    public static int sum(int a, int b) {  
        return a + b;  
    }  
  
    public static void main(String[] args) {  
        // Пример использования функции  
        int result = sum(5, 3);  
        System.out.println("Сумма: " + result); // Выведет: Сумма: 8  
    }  
}
```

5.6. Генерация unit-тестов

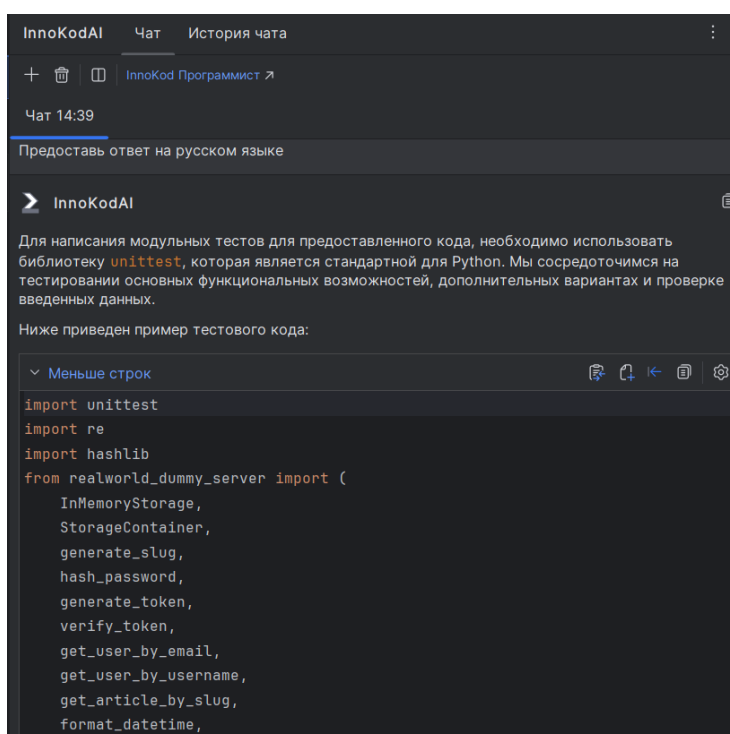
Плагин для интегрированной среды разработки позволяет сгенерировать unit-тесты для выбранной Пользователем части программного кода.

Порядок использования:

- В IDE IntelliJ IDEA открыть файл проекта, который содержит программный код, для которого необходимо сгенерировать unit-тест;
- В открытом файле выделить часть кода, для которой необходимо сгенерировать unit-тест, нажать правую кнопку мыши;
- В открывшемся списке меню выбрать «InnoKod»;
- В открывшемся списке выбрать «Написать тесты»;



- После выбора данного пункта меню unit-тест будет сгенерирован в виде ответа в чате (Ниже приведен отрывок ответа с результатом подготовленного теста)

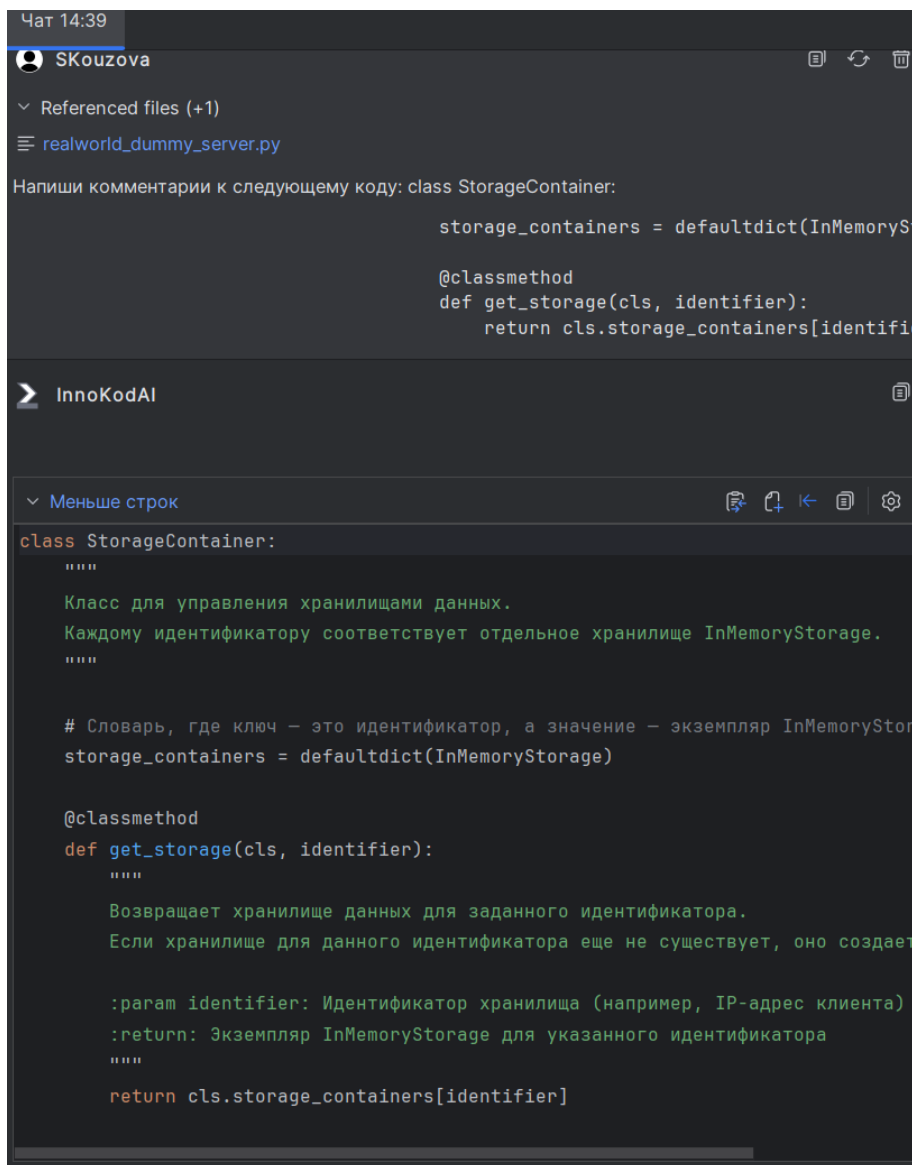


5.7. Генерация комментариев

Плагин позволяет автоматически генерировать комментарии к выделенному программному коду.

Для запуска данной функции необходимо:

- Открыть файл с программным кодом в IDE IntelliJ IDEA;
- Выделить часть кода, к которой необходимо сгенерировать комментарий, скопировать его;
- Написать сообщение с текстом «Напиши комментарии к следующему коду» в область написания вопросов в ИИ-чате и добавить код, скопированный на предыдущем шаге.
- Нажать кнопку Enter. После отправки вопроса в чате будет отображен ответ с предложением комментариев к коду.



Чат 14:39

SKouzova

Referenced files (+1)

realworld_dummy_server.py

Напиши комментарии к следующему коду: class StorageContainer:

```
storage_containers = defaultdict(InMemoryStorage)

@classmethod
def get_storage(cls, identifier):
    return cls.storage_containers[identifier]
```

InnoKodAI

Меньше строк

```
class StorageContainer:
    """
    Класс для управления хранилищами данных.
    Каждому идентификатору соответствует отдельное хранилище InMemoryStorage.
    """

    # Словарь, где ключ — это идентификатор, а значение — экземпляр InMemoryStorage
    storage_containers = defaultdict(InMemoryStorage)

    @classmethod
    def get_storage(cls, identifier):
        """
        Возвращает хранилище данных для заданного идентификатора.
        Если хранилище для данного идентификатора еще не существует, оно создается.

        :param identifier: Идентификатор хранилища (например, IP-адрес клиента)
        :return: Экземпляр InMemoryStorage для указанного идентификатора
        """

        return cls.storage_containers[identifier]
```

5.8. Автодополнение кода

Плагин для интегрированной среды разработки выполняет автоматическое дополнение программного кода при написании кода в файле, открытом в IDE.

Пример работы:



```
28     public static int factorial(int i) { no usages new *
29         if (i == 0) {
            return 1;
        } else {
            return i * factorial(i - 1);
        }
30     }
31
32 }
```