Продукт Platform V API Mock & Contract Testing (AMC) Компонент Сервис умных заглушек и контрактного тестирования (SberMock) (AMCS)

Описание функциональных характеристик

Содержание

Описание функциональных характеристик продукта Platform V API Mock & Contract Testing (AMC)

Термины и определения

Термин	Определение
БД	База данных
ПО	Программное обеспечение
Платформа	Набор продуктов Platform V, правообладателем которых является АО «СберТех». Перечень таких продуктов обозначен в документации на конкретный продукт
СУБД	Система управления базами данных
ФП	Функциональная подсистема
Эмуляция	Воспроизведение программными или аппаратными средствами (или их комбинацией) работы других программ или устройств
AMCS	Компонент Сервис умных заглушек и контрактного тестирования (SberMock) в составе продукта Platform V API Mock & Contract Testing (AMC)
API	Application Programming Interface — программный интерфейс приложения, описывающий способы взаимодействия с другими программами
Eureka	Приложение для поиска необходимых микросервисов
HTTP	HyperText Transfer Protocol («протокол передачи гипертекста») — протокол прикладного уровня передачи данных
HTTPs	HyperText Transfer Protocol Secure — расширение протокола HTTP для поддержки шифрования в целях повышения безопасности
TLS	Transport Layer Security — протокол защиты транспортного уровня
URL	Uniform Resource Locator — адрес ресурса в интернете

Цель создания

Сервис интеграционного тестирования API на контрактах и заглушках позволяет генерировать интеграционные заглушки API и вести их реестр для ускорения разработки приложений. Также данный сервис помогает повысить качества кода за счет раннего тестирования интеграционных взаимодействий.

Основные функции

Название функции	Потребитель функции	Аргументы	Результат
Использование проектной области для создания эмуляций API	1. Пользователь 2. Группа пользователей	1. **Имя** и **Описание** проектной области 2. Данные для эмуляции **Базовый url** и **Хост**	Отделение проектов пользователей друг от друга, привязка пользователя или группы пользователей (команды) к Проектной области.
Настройка REST-эмуляции	1. Клиентское приложение 2. Тестируемый функционал	1. Наименование эмуляции 2. URL запроса 3. Параметры HTTP (код ответа, заголовок, тело)	Настроенная REST-эмуляция для тестирования интеграционного взаимодействия с клиентским приложением.

Название функции	Потребитель функции	Аргументы	Результат
		4. Шаблон REST-эмуляции (Groovy, JSON, XML)	
Настройка области переменных	1. Клиентское приложение 2. Тестируемый функционал	1. Наименование области переменных 2. Имя переменной и ее тип 3. Значение переменой по умолчанию	Область переменных доступна для использования в REST-эмуляции.
Просмотр истории вызовов эмуляции	Пользователь	Наименование эмуляции	В интерфейсе компонента AMCS открывается просмотр истории и результатов вызова эмуляций в рамках проектной области.
Просмотр публичного реестра для работы с эмуляциями	1. Пользователь 2. Клиентское приложение 3. Тестируемый функционал	Наименование эмуляций, помеченных своими владельцами как публичные	Эмуляции доступны на просмотр и использование любому Пользователю компонента AMCS. Эмуляции также доступны коллегам, не имеющим доступа к Проектной области Пользователя.

Сценарии использования

Вариант использования	Пользователь передает данные в компонент	Компонент передает данные пользователю
Синхронный ответ REST-эмуляции	Клиентское приложение делает HTTP-запрос в REST-эмуляцию	REST-эмуляция сразу же возвращает HTTP-ответ
Statefull синхронный ответ REST-эмуляции	Клиентское приложение делает HTTP-запрос в REST-эмуляцию	1. REST-эмуляция вызывает Groovy скрипт, который формирует HTTP-ответ на основе данных, хранимых в Области. 2. REST-эмуляция возвращает HTTP-ответ клиентскому приложению.
Асинхронный ответ REST-эмуляции	1. Клиентское приложение делает HTTP-запрос в REST-эмуляцию. 2. Клиентское приложение ожидает асинхронного ответа. 3. Клиентское приложение отправляет HTTP-ответ.	1. REST-эмуляция сразу же возвращает HTTP-ответ. 2. REST-эмуляция по таймауту инициирует вызов асинхронного ответа. 3. REST-эмуляция делает HTTP-запрос клиентскому приложению.