Описание функциональных характеристик

Продукта Platform V Batch (BAT)

ОГЛАВЛЕНИЕ

Описание функциональных характеристики компонента Пакетная обработка задач (Batch Tasks)

Назначение

Цель создания

Компонент Пакетная обработка задач (Batch Tasks) продукта Platform V Batch (далее — сервис Batch Tasks, Пакетная обработка задач) (TASK) предназначен для асинхронного запуска вычислительных задач в порядке очереди. Сервис Batch Tasks предоставляет возможность управлять простыми операциями и контролировать результаты их выполнения с целью автоматизации фоновых бизнес-процессов.

Основными преимуществами сервиса являются:

- использование протокола HTTP(S) для быстрого и безопасного обмена данными;
- единый интерфейс для администрирования Задач и Очередей: создание, редактирование, удаление, контроль выполнения;
- свобода потребителя в схеме размещения;
- свобода потребителя в выборе языка реализации;
- работа без подключения сторонних библиотек;
- возможность задания уникальной политики повторов Задачи для каждой Очереди;
- установка ограничения на максимальное количество одновременно выполняемых Задач в Очереди;
- возможность установки зависимости между Задачами – успешное выполнение Задачи влияет на запуск других Задач.

Основные функции

Основной функцией сервиса Batch Tasks является выполнение Задач в порядке Очереди в соответствии с заданными параметрами.

В таблице ниже приведены дополнительные возможности для настройки параметров сервиса:

№	Название функции	Потребитель	Аргументы	Результат
1	Выполнение Задач в Очереди	Потребитель	Параметры Очередей и Задач	Задача выполняется в соответствии с заданными параметрами
2	Объединение Очередей в группы	Потребитель	Параметры Очередей и групп	Настройки группировки применяются

3	Зависимости Задач	Потребитель	Настройки зависимостей	Задачи выполняются в соответствии с настройками
4	Политика повтора Задач	Потребитель	Настройки политики повтора	Задачи выполняются в соответствии с политиками повтора
5	Отложенный запуск Задач	Потребитель	Настройки планируемого запуска Задач	Задачи выполняются в соответствии с настройками

Сценарии использования

Сценарий 1. Выполнение Задач в Очереди

Данный сценарий описывает выполнение Задач в Очереди.

1. Пользователь создает Очередь посредством UI или вызовом метода API CreateQueue. Пользователь становится владельцем созданного экземпляра Очереди.
2. Пользователь создает Задачу посредством UI или вызовом метода API CreateTask, со следующими свойствами:
   - в атрибуте queue указывается Очередь, для которой выполняется Задача;
   - в атрибуте httpTarget указывается вызов пользовательского веб-сервиса.
3. Сервис начинает брать Задачи из Очереди и выполнять их. Во избежание возникновения race conditions в пользовательском веб-сервисе, ответственном за проверку наличия новых порций данных для обработки и постановку новых Задач в Очередь, рекомендуется установить лимит параллельного запуска Задач в Очереди в единицу, чтобы Задачи запускались последовательно.

Сценарий 2. Выполнение регулярных объемных вычислений

Объемные регулярные вычисления – это вычисления, объем работы в которых выполняется большим количеством фиксированных Задач. Так как выполнение одного Задания более 15 мин прерывается, то даже разделение одного большого Задания на несколько маленьких не является оптимальным способом, если необходимо создать несколько десятков, сотен или более Заданий. Для реализации этого сценария существует механизм Очередей и Задач.

1. Пользователь создает регулярно запускаемое Задание посредством UI или вызовом метода API CreateJob, со следующими свойствами:
   - в атрибуте schedule задается расписание запуска Задания в cron-формате;
   - в атрибуте httpTarget задается объект HttpTarget, описывающий запускаемое вычисление;
   - данный пользователь становится владельцем созданного экземпляра Задания.
2. HttpTarget в Задании должен указывать на веб-сервис пользователя, вызов которого:
   - создаст Очередь вызовом метода API Планировщика CreateQueue;
   - разделит вычисление на порции (диапазоны обрабатываемых данных);
   - создаст в очереди Задачи вызовами метода API Планировщика CreateTask с соответствующим HttpTarget для обработки определенной порции данных;
   - опционально, добавит в Очереди Задачу с вызовом метода API Планировщика DeleteQueue для удаления этой Очереди. Данный прием позволит удалить Очередь после выполнения последнего вычисления пользователя.
3. Планировщик выполнит действия в следующем порядке:
   - вызов Задания по расписанию с веб-сервисом пользователя, который создает Очередь и Задачи в Очереди;
   - выполняет Задачи с начала Очереди, обработав вычисления пользователя;
   - вызов Задачи для удаления самой Очереди.

Сценарий 3. Пополняемые Очереди Задач

Данный сценарий описывает возможность пополнения Очереди в процессе выполнения Задач.

1. Пользователь создает Очередь посредством UI или вызовом метода API CreateQueue. Пользователь становится владельцем созданного экземпляра Очереди.
2. Пользователь создает Задание посредством UI или вызовом метода API Create со следующими свойствами:
   - в атрибуте **Расписание запуска** / schedule задается расписание запуска Задания в cron-формате;
   - в атрибуте httpTarget должен быть вызов пользовательского веб-сервиса, который проверяет наличие новых вычислительных Задач и добавляет их в Очередь, созданную в пункте 1. Наименование Очереди может быть задано в HttpTarget Задания;
   - пользователь становится владельцем созданного экземпляра Задания.
3. Вычислительные веб-сервисы, вызываемыми Задачами, также должны после завершения вычисления, проверять наличие новых вычислительных Задач и добавлять их в Очередь созданную в п.1. Наименование Очереди может быть задано в HttpTarget Задачи.
4. Планировщик запустит Задание согласно расписанию. Пользовательский веб-сервис, вызванный Заданием, создаст Задачи в Очереди.

5. Планировщик начнет брать Задачи из Очереди и выполнять их. Пользовательский веб-сервис, вызываемый в Задачах (Task), будет проверять наличие новых порций пользовательских данных для обработки, и добавлять их в Очередь в виде Задач, при необходимости. Во избежание возникновения race conditions в пользовательском веб-сервисе, ответственном за проверку наличия новых порций данных для обработки и постановку новых Задач в Очередь, рекомендуется установить лимит параллельного запуска Задач в Очереди в 1 (единицу), чтобы Задачи запускались последовательно.
6. Если Задачи в Очереди закончатся и новых Задач в процессе выполнения Задач не будет создано, выполнение Задач закончится. Но регулярный запуск Задания позволит возобновить обработку вычислений.

Описание функциональных характеристики компонента Планировщик заданий (Batch Scheduler)

Назначение

Цель создания

Компонент Планировщик заданий (Batch Scheduler) продукта Platform V Batch (SCHD) (далее — сервис Batch Scheduler, Планировщик заданий) предназначен для запуска вычислительных заданий по расписанию или по требованию (запрос API и через UI). Вычислительные Задания оформлены в виде вызова произвольных запросов, которые вызывают веб-сервисы по протоколу HTTP(S), включая архитектурный стиль REST и протокол JSON-RPC 2.0. Сервис Batch Scheduler предоставляет возможность управлять простыми операциями и контролировать результаты их выполнения с целью автоматизации фоновых бизнес-процессов.

Основными преимуществами сервиса являются:

- использование протокола HTTP(S) для быстрого и безопасного обмена данными;
- единый интерфейс для администрирования Заданий: создание, редактирование, удаление, контроль выполнения;
- свобода потребителя в схеме размещения;
- свобода потребителя в выборе языка реализации;
- работа без подключения сторонних библиотек;
- автоматическая обработка сбоев — сбой при выполнении отдельного вызова не окажет существенного влияния на приложение, если задание запускается каждые 15—20 мин и обрабатывает все запросы;
- запуск Задания по запросу API с надежностью — 99,99 %;
- поддержка cron-формата.

Основные функции

Основными функциями сервиса Batch Scheduler является запуск задания по расписанию или по требованию (запрос API и через UI).

В таблице ниже приведены варианты запуска вычислений.

№	Наименование	Потребитель	Аргументы	Результат
1	Запуск Задания по расписанию	Потребитель	Параметры задания	Планировщик регулярно запускает Задания в заданное время согласно расписанию в cron-формате
2	Запуск Задания по требованию (запрос API и через UI)	Потребитель	Параметры задания	Планировщик запускает Задания по запросам пользователей к API Планировщика
3	Политика повтора Заданий	Потребитель	Настройки политики повтора	Задания выполняются в соответствии с политиками повтора

Сценарии использования

Сценарий 1. Регулярный запуск вычисления по расписанию

Для регулярного запуска вычислений по расписанию необходимо использовать механизм Заданий согласно следующему сценарию:

1. Пользователь создает Задание посредством UI или вызовом метода API create со следующими свойствами:
   - в атрибуте **Расписание запуска** / schedule задается расписание запуска Задания в cron-формате;
   - в атрибутах **URL адрес**, **Список HTTP заголовков**, **Метод** / httpTarget задается объект HttpTarget, описывающий запускаемое вычисление;
   - пользователь становится владельцем созданного экземпляра Задания.
2. Планировщик заданий регулярно делает запросы HTTP(S) по URL, заданному в объекте HttpTarget (сущность компонента Batch Scheduler (SCHD)), согласно расписанию schedule. Вызов производится синхронно, и Планировщик Заданий ожидает выполнения запроса и возврата ответа.
3. Веб-сервис, вызываемый через HttpTarget, может изменять атрибуты Задания через UI или вызовом метода API update.

Если время выполнения Задания превышает время выполнения HttpTarget (15 мин), то можно создать несколько одинаковых Заданий, отличающиеся входными параметрами вызываемого веб-сервиса, для разделения одного объемного вычисления на несколько небольших.

Сценарий 2. Запуск повторяющегося вычисления по требованию

Механизм позволяет реализовать запуск повторяющихся вычислений по требованию — через UI или с помощью запроса к API:

1. Пользователь создает Задание посредством UI или вызовом метода API create со следующими свойствами:
   - атрибут **Расписание запуска** / schedule должен быть пустой строкой "", либо отсутствовать в запросе. Таким образом, расписание Задания отсутствует, и запуск Задания возможен только по запросу;
   - в атрибуте httpTarget задается объект HttpTarget, описывающий запускаемое вычисление;
   - пользователь становится владельцем созданного экземпляра Задания.
2. Пользователь вызывает метод API run с наименованием Задания, либо запускает Задание посредством UI. Метод run возвращает результат пользователю немедленно, при этом HttpTarget в составе задания выполняется асинхронно от вызова API.
3. Планировщик вызывает HttpTarget независимо от принятого запроса. Вызов URL в HttpTarget производится синхронно, Планировщик заданий ожидает выполнения запроса и возврата ответа от веб-сервиса, заданного в HttpTarget. Время выполнения запроса не должно превышать времени ожидания HttpTarget, то есть 15 мин.

Примечание. Между запусками повторяющегося вычисления проходит не менее 1 мин.

Сценарий 3. Выполнение регулярных объемных вычислений

Объемные регулярные вычисления – это вычисления, объем работы которых выполняется большим количеством фиксированных Заданий. Если необходимо создать несколько десятков, сотен или более Заданий, то для реализации этого сценария существует механизм Очередей и Задач. Выполнение одного Задания длительностью более 15 мин прерывается, поэтому при проведении объемных вычислений разделение одного большого Задания на несколько частей не является оптимальным способом.

1. Пользователь создает регулярно запускаемое Задание посредством UI или вызовом метода API create со следующими свойствами:
   - в атрибуте schedule задается расписание запуска Задания в cron-формате;
   - в атрибуте httpTarget задается объект HttpTarget, описывающий запускаемое вычисление;
   - пользователь становится владельцем созданного экземпляра Задания.
2. HttpTarget в Задании должен указывать на веб-сервис пользователя, вызов которого:
   - создаст Очередь вызовом API Планировщика create;
   - разделит вычисление на порции (диапазоны обрабатываемых данных);
   - создаст в Очереди Задачи вызовами API Планировщика create с соответствующим HttpTarget для обработки определенной порции данных;
   - опционально, добавит в Очереди Задачу с вызовом API Планировщика delete для удаления этой Очереди. Данный прием позволит удалить очередь после выполнения последнего вычисления пользователя.
3. Планировщик выполнит действия в следующем порядке:
   - вызов Задания по расписанию с веб-сервисом пользователя, который создаст Очередь и Задачи в Очереди;
   - выполнение Задачи с начала Очереди, обработав вычисления пользователя;
   - вызов Задачи для удаления самой Очереди.

Сценарий 4. Пополняемые Очереди Задач

Данный сценарий описывает возможность пополнения Очереди в процессе выполнения Задач.

1. Пользователь создает Очередь посредством UI или вызовом метода API create. Пользователь становится владельцем созданного экземпляра Очереди.
2. Пользователь создает Задание посредством UI или вызовом метода API create, со следующими свойствами:
   - в атрибуте **Расписание запуска** / schedule задается расписание запуска Задания в cron-формате;
   - в атрибуте httpTarget должен быть вызов пользовательского веб-сервиса, который проверяет наличие новых вычислительных Задач и добавляет их в Очередь, созданную в пункте 1. Наименование Очереди может быть задано в HttpTarget Задания;
   - пользователь становится владельцем созданного экземпляра Задания.

3. Вычислительные веб-сервисы, вызываемыми Задачами, также должны после завершения вычисления, проверять наличие новых вычислительных Задач и добавлять их в Очередь созданную в пункте 1. Наименование Очереди может быть задано в HttpTarget Задачи.
4. Планировщик запустит Задание согласно расписанию. Пользовательский веб-сервис, вызванный Заданием, создаст Задачи в Очереди.
5. Планировщик начнет брать Задачи из Очереди и выполнять их. Пользовательский веб-сервис, вызываемый в Задачах, будет проверять наличие новых порций пользовательских данных для обработки, и добавлять их в Очередь в виде Задач, при необходимости. Во избежание возникновения race conditions в пользовательском веб-сервисе, ответственном за проверку наличия новых порций данных для обработки и постановку новых Задач в Очередь, рекомендуется установить лимит параллельного запуска Задач в Очереди в 1 (единицу), чтобы Задачи запускались последовательно.
6. Если Задачи в Очереди закончатся и новых Задач в процессе выполнения Задач не будет создано, выполнение Задач закончится. Но регулярный запуск Задания позволит возобновить обработку вычислений.