|  |  |
| --- | --- |
| Министерство образования и науки Российской Федерации  Федеральное государственное бюджетное образовательное учреждение  высшего профессионального образования  «Ижевский государственный технический университет  имени М. Т. Калашникова» | |
| Кафедра «Программное обеспечение» | |
| Отчёт по лабораторной работе №2  дисциплины «КПО» | |
| Выполнил:  Студент группы Б08-191-2 | Р. С. Поскребышев |
| Принял: |  |
| Ижевск 2016 | |

Предварительные требования

1. Bacs API должен содержать методы:
   1. Получить множество задач
      1. Описание: метод должен возвращать список задач по заданному идентификатору контеста.
      2. Входные данные:
         1. Идентификатор контеста
            1. Тип: целое число
      3. Выходные данные:
         1. Список задач
            1. Тип: список объектов типа Problem
   2. Отправить решение на проверку
      1. Описание: метод должен отсылать на проверку входное решение и выдавать идентификатор результата проверки.
      2. Входные данные:
         1. Решение задачи
            1. Тип: Submit
      3. Выходные данные:
         1. Идентификатор результата решения
            1. Тип: целое число
   3. Отправить множество решений на проверку
      1. Описание: метод должен отсылать на проверку множество решений и выдавать в соответствующем порядке множество результатов.
      2. Входные данные:
         1. Список решений задач
            1. Тип: список объектов типа Submit
      3. Выходные данные:
         1. Список идентификаторов результатов
            1. Тип: список целых чисел
   4. Перепроверить решение
      1. Описание: метод должен отправлять уже имеющееся в базе решение заново на проверку.
      2. Входные данные:
         1. Идентификатор решения
            1. Тип: целое число
      3. Выходные данные:
         1. Идентификатор результата
            1. Тип: целое число
   5. Перепроверить множество решений
      1. Описание: метод должен переотправлять все заданные решения, которые уже имеются в базе, заново на проверку.
      2. Входные данные:
         1. Список идентификатором решений
            1. Тип: список целых чисел
      3. Выходные данные:
         1. Список идентификаторов результатов
            1. Тип: список целых чисел
   6. Получить результат решения
      1. Описание: метод должен возвращать результат решения по заданному идентификатору.
      2. Входные данные:
         1. Идентификатор решения
            1. Тип: целое число
      3. Выходные данные:
         1. Результат решения
            1. Тип: класс SubmitResult
   7. Получить множество результатов
      1. Описание: метод должен соответствующий возвращать список результатов по заданному списку идентификаторов этих результатов.
      2. Входные данные:
         1. Список идентификаторов результатов:
            1. Тип: список целых чисел
      3. Выходные данные
         1. Список результатов решений
            1. Тип: список объектов типа SubmitResult
   8. Получить ссылку на условие задачи
      1. Описание: метод должен возвращать ссылку на условие задачи по заданному идентификатору контеста и идентификатору задачи в контесте.
      2. Входные данные:
         1. Идентификатор контеста
            1. Тип: целое число
         2. Идентификатор задачи в контесте
            1. Тип: строковое значение
      3. Выходные данные:
         1. Ссылка на условие задачи
            1. Тип: строковое значение
   9. Получить список поддерживаемых языков программирования
      1. Описание: метод должен возвращать список языков программирования
      2. Входных данных нет.
      3. Выходные данные:
         1. Список языков программирования
            1. Тип: список строковых объектов
2. Программная реализация продукта должна использовать готовые системы:
   1. Система тестирования Bacs.
   2. Система авторизации и аутентификации Bacs.
   3. Систему ролей Bacs.
   4. Система управления архивом Bacs.
   5. ORM Entity Framework.
   6. База данных Bacs.
3. Интерфейс спецификации
   1. Интерфейс спецификации должен генерироваться с помощью готового инструмента Swagger.
   2. Интерфейс для каждого метода должен содержать:
      1. Описание входных данных.
      2. Описание выходных данных.
      3. Пример составления запроса.
      4. Пример составления ответа.
      5. Возможность отправить свой запрос прямо из графического интерфейса.
      6. После отправки своего запроса должен появляться ответ запроса.

Вопросы

1. Как происходит авторизация на стороне API?
2. Какие методы доступны каждой роли?
3. Какие поля доступны каждой роли?
4. Откуда гость возьмет идентификаторы результатов?
5. Что такое контест? Нужно переименовать для понимания более широкого круга потребителей.
6. Открытые контесты не нужны.
7. Нужен ли гость?
8. Как получить идентификатор языка программирования?
9. Задача должна иметь свой уникальный идентификатор типа Int64.
10. Заменить понятие «целое число» на «Int64».

Обновленные требования

Archive API

1. Авторизация на стороне API происходит по токену, который выдается ЦАА (Центр Авторизации и Аутентификации).
2. Роли устанавливаются ЦАА:
   1. Гость (*Guest*);
   2. Зарегистрированный пользователь (*RegistratedUser*);
   3. Подтвержденный пользователь (*ConfirmedUser*);
   4. Супер пользователь (*SuperUser*).
3. Методы API:

|  |  |  |
| --- | --- | --- |
| Название метода | | |
| Описание метода | Входные данные | Выходные данные |
| Получить множество задач (*GetProblems*) | | |
| Метод должен возвращать список задач по заданному идентификатору коллекции. | Идентификатор коллекции  Тип: Int64 | Список задач  Тип: список объектов типа Problem |
| Получить ссылку на условие задачи (*GetStatementUrl*) | | |
| Метод должен возвращать ссылку на условие задачи по заданному идентификатору задачи. | Идентификатор задачи  Тип: Int64 | Ссылка на условие задачи  Тип: строковое значение |

1. Ограничения по ролям для методов:

|  |  |  |
| --- | --- | --- |
| Метод | Роли | Ограничения |
| Получить множество задач (*GetProblems*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Задачи из доступной только ему коллекции |
| Супер пользователь (*SuperUser*) | Полный доступ |
| Получить ссылку на условие задачи (*GetStatementUrl*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Задача из доступной только ему коллекции |
| Супер пользователь (*SuperUser*) | Полный доступ |

1. Типы:

|  |  |  |
| --- | --- | --- |
| Название типа | | |
| Название поля | Тип поля | Описание поля |
| *Problem* | | |
| *Info* | *Map<String, ProblemInfo>* | Информация по задаче для каждого языка |
| *Statement* | *Statement* | Условие |
| *ResourceLimits* | *ProblemResourceLimits* | Ограничения на ресурсы |
| *Statement* | | |
| *Version* | *StatementVersion* | Версия условия |
| *StatementVersion* | | |
| *Language* | *String* | Язык, на котором написано условие |
| *Format* | *String* | Расширение файла |
| *ProblemInfo* | | |
| *Name* | *String* | Название задачи |
| *ProblemResourceLimits* | | |
| *TimeLimitMillis* | *Int64* | Ограничение по времени в миллисекундах |
| *MemoryLimitBytes* | *Int64* | Ограничение по памяти в байтах |

1. Ограничения по ролям для типов:

|  |  |  |
| --- | --- | --- |
| Название типа | | |
| Название поля | Роли | Ограничения |
| *Problem* | | |
| *Info* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *Statement* | Гость (*Guest*) | Полный доступ |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *ResourceLimits* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *Statement* | | |
| *Version* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *StatementVersion* | | |
| *Language* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *Format* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *ProblemInfo* | | |
| *Name* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *ProblemResourceLimits* | | |
| *TimeLimitMillis* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *MemoryLimitBytes* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Полный доступ |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |

1. Программная реализация продукта должна использовать готовые системы:
   1. Система тестирования (*Executor API*);
   2. Система авторизации и аутентификации (*CAA API*);
   3. Систему ролей *CAA*;
   4. *ORM Entity Framework*.
2. Интерфейс спецификации
   1. Интерфейс спецификации должен генерироваться с помощью готового инструмента Swagger.
   2. Интерфейс для каждого метода должен содержать:
      1. Описание входных данных.
      2. Описание выходных данных.
      3. Пример составления запроса.
      4. Пример составления ответа.
      5. Возможность отправить свой запрос прямо из графического интерфейса.
      6. После отправки своего запроса должен появляться ответ запроса.

Executor API

1. Авторизация на стороне API происходит по токену, который выдается ЦАА (Центр Авторизации и Аутентификации).
2. Роли устанавливаются ЦАА:
   1. Гость (*Guest*);
   2. Зарегистрированный пользователь (*RegistratedUser*);
   3. Подтвержденный пользователь (*ConfirmedUser*);
   4. Супер пользователь (*SuperUser*).
3. Executor API должен содержать методы:

|  |  |  |
| --- | --- | --- |
| Название метода | | |
| Описание метода | Входные данные | Выходные данные |
| Отправить множество решений на проверку (*SendSubmits*) | | |
| Метод должен отсылать на проверку множество решений и выдавать в соответствующем порядке множество результатов. | Список решений задач  Тип: список объектов типа Submit | Список идентификаторов результатов  Тип: список целых чисел |
| Перепроверить множество решений (*RejudgeSubmits*) | | |
| Метод должен переотправлять все заданные решения, которые уже имеются в базе, заново на проверку. | Список идентификатором решений  Тип: список целых чисел | Список идентификаторов результатов  Тип: список целых чисел |
| Получить множество результатов (*GetSubmitResults*) | | |
| Метод должен соответствующий возвращать список результатов по заданному списку идентификаторов этих результатов. | Список идентификаторов результатов  Тип: список целых чисел | Список результатов решений  Тип: список объектов типа SubmitResult |
| Получить список поддерживаемых языков программирования (*GetLangs*) | | |
| Метод должен возвращать список языков программирования. | - | Список языков программирования  Тип: список объектов типа Language |

1. Ограничение по ролям для методов:

|  |  |  |
| --- | --- | --- |
| Метод | Роли | Ограничения |
| Отправить множество решений на проверку (*SendSubmits*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Проверка решений задач из доступной только ему коллекции |
| Супер пользователь (*SuperUser*) | Полный доступ |
| Перепроверить множество решений (*RejudgeSubmits*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Проверка решений задач из доступной только ему коллекции |
| Супер пользователь (*SuperUser*) | Полный доступ |
| Получить множество результатов (*GetSubmitResults*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Открытая информация результатов (*SubmitResult*) по задачам из доступных ему коллекций, либо доступная информация результатов по своим посылкам (*Submit*) |
| Супер пользователь (*SuperUser*) | Полный доступ |
| Получить список поддерживаемых языков программирования (*GetLangs*) | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |

1. Типы:

|  |  |  |
| --- | --- | --- |
| Название типа | | |
| Название поля | Тип поля | Описание поля |
| *Submit* | | |
| *Id* | *Int64* | Идентификатор посылки |
| *LangId* | *Int64* | Идентификатор языка программирования |
| *Solution* | *Byte[]* | Решение |
| *SubmitResult* | | |
| *Id* | *Int64* | Идентификатор результата посылки |
| *BuildResult* | *BuildResult* | Результат построения решения |
| *TestResults* | *TestGroupResult[]* |  |
| *BuildResult* | | |
| *Status* | *enum BuildResultStatus* | Результат построения решения |
| *Output* | *String* | Вывод компилятора |
| *TestGroupResult* | | |
| *Executed* | *Boolean* | Было ли проверено решение на группе тестов |
| *TestResults* | *TestResult[]* | Список результатов по каждому тесту |
| *TestResult* | | |
| *Status* | *enum JudgeResultStatus* | Результат проверки решения на тесте |
| *JudgeMessage* | *String* | Сообщение проверяющей системы |
| *ResourceUsage* | *ResourceUsage* | Затраченные ресурсы |
| *ResourceUsage* | | |
| *TimeUsageMillis* | *Int64* | Затраченное время в миллисекундах |
| *MemoryUsageBytes* | *Int64* | Затраченная память в байтах |
| *Language* | | |
| *Id* | *Int64* | Идентификатор языка программирования |
| *Name* | *String* | Название языка программирования |

1. Перечисления (enums):

|  |  |  |
| --- | --- | --- |
| Название перечисления | | |
| Элементы перечисления | Код (значение) | Описание |
| *BuildResultStatus* | | |
| *Ok* | 0 | Построение завершено успешно |
| *Failed* | 1 | Завершение завершилось с ошибкой |
| *JudgeResultStatus* | | |
| *Ok* | 0 | Тест пройден |
| *WrongAnswer* | 1 | Неверный ответ на тесте |
| *PresentationError* | 2 | Неверный формат вывода |
| *QueriesLimitExceeded* | 3 | Превышено количество запросов |
| *IncorrectRequest* | 4 | Неверный формат запроса |
| *InsufficientData* | 5 |  |
| *ExcessData* | 6 |  |
| *OutputLimitExceeded* | 7 | Превышено ограничение на вывод |
| *TerminationRealTimeLimitExceeded* | 8 |  |
| *CustomFailure* | 500 |  |
| *FailTest* | 999 |  |
| *Failed* | 1000 |  |
| *Skipped* | 2000 | Тест пропущен |

1. Ограничения по ролям для типов:

|  |  |  |
| --- | --- | --- |
| Название типа | | |
| Название поля | Роли | Ограничения |
| *Submit* | | |
| *Id* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Нет доступа |
| Супер пользователь (*SuperUser*) | Нет доступа |
| *LangId* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Нет доступа |
| Супер пользователь (*SuperUser*) | Нет доступа |
| *Solution* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Нет доступа |
| Супер пользователь (*SuperUser*) | Нет доступа |
| *SubmitResult* | | |
| *Id* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *BuildResult* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *TestResults* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *BuildResult* | | |
| *Status* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *Output* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *TestGroupResult* | | |
| *Executed* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *TestResults* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *TestResult* | | |
| *Status* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *JudgeMessage* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *ResourceUsage* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *ResourceUsage* | | |
| *TimeUsageMillis* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |
| *MemoryUsageBytes* | Гость (*Guest*) | Нет доступа |
| Зарегистрированный пользователь (*RegistratedUser*) | Нет доступа |
| Подтвержденный пользователь (*ConfirmedUser*) | Полный доступ |
| Супер пользователь (*SuperUser*) | Полный доступ |

1. Программная реализация продукта должна использовать готовые системы:
   1. Система тестирования (*Bunsan API*);
   2. Система авторизации и аутентификации (*CAA API*);
   3. Систему ролей *CAA*;
   4. Система управления архивом (*Archive API*);
   5. *ORM Entity Framework*.
2. Интерфейс спецификации
   1. Интерфейс спецификации должен генерироваться с помощью готового инструмента Swagger.
   2. Интерфейс для каждого метода должен содержать:
      1. Описание входных данных.
      2. Описание выходных данных.
      3. Пример составления запроса.
      4. Пример составления ответа.
      5. Возможность отправить свой запрос прямо из графического интерфейса.
      6. После отправки своего запроса должен появляться ответ запроса.