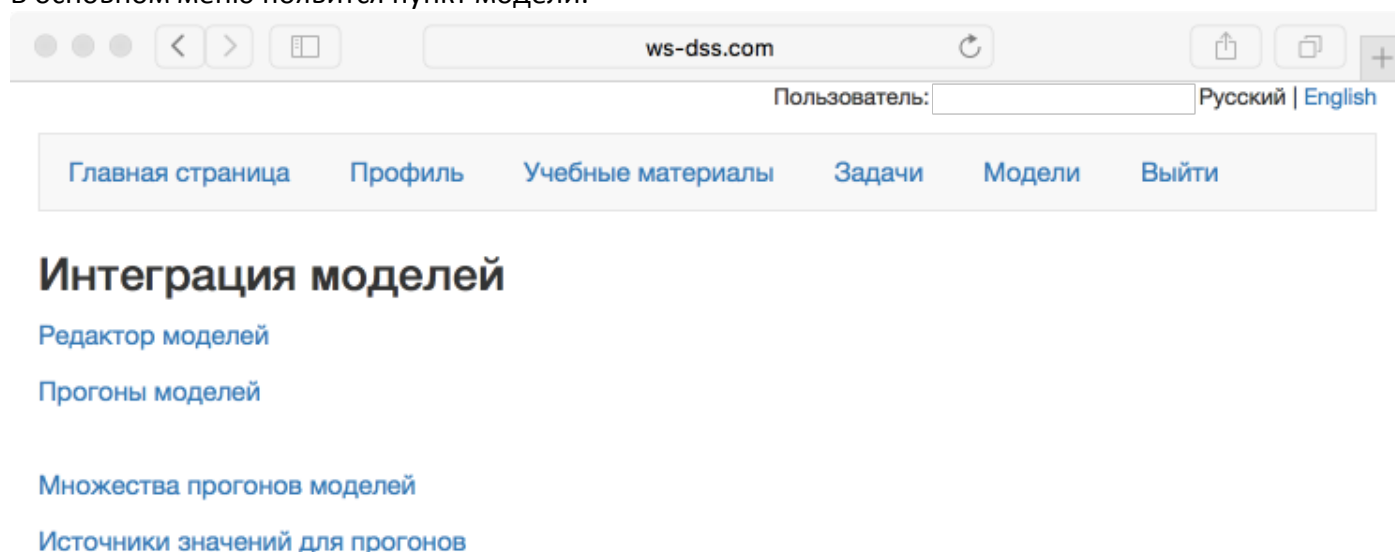


Руководство по использованию моделей в ws-dss.com

Использование моделей из интерфейса пользователя

Зарегистрируйтесь на сайте ws-dss.com

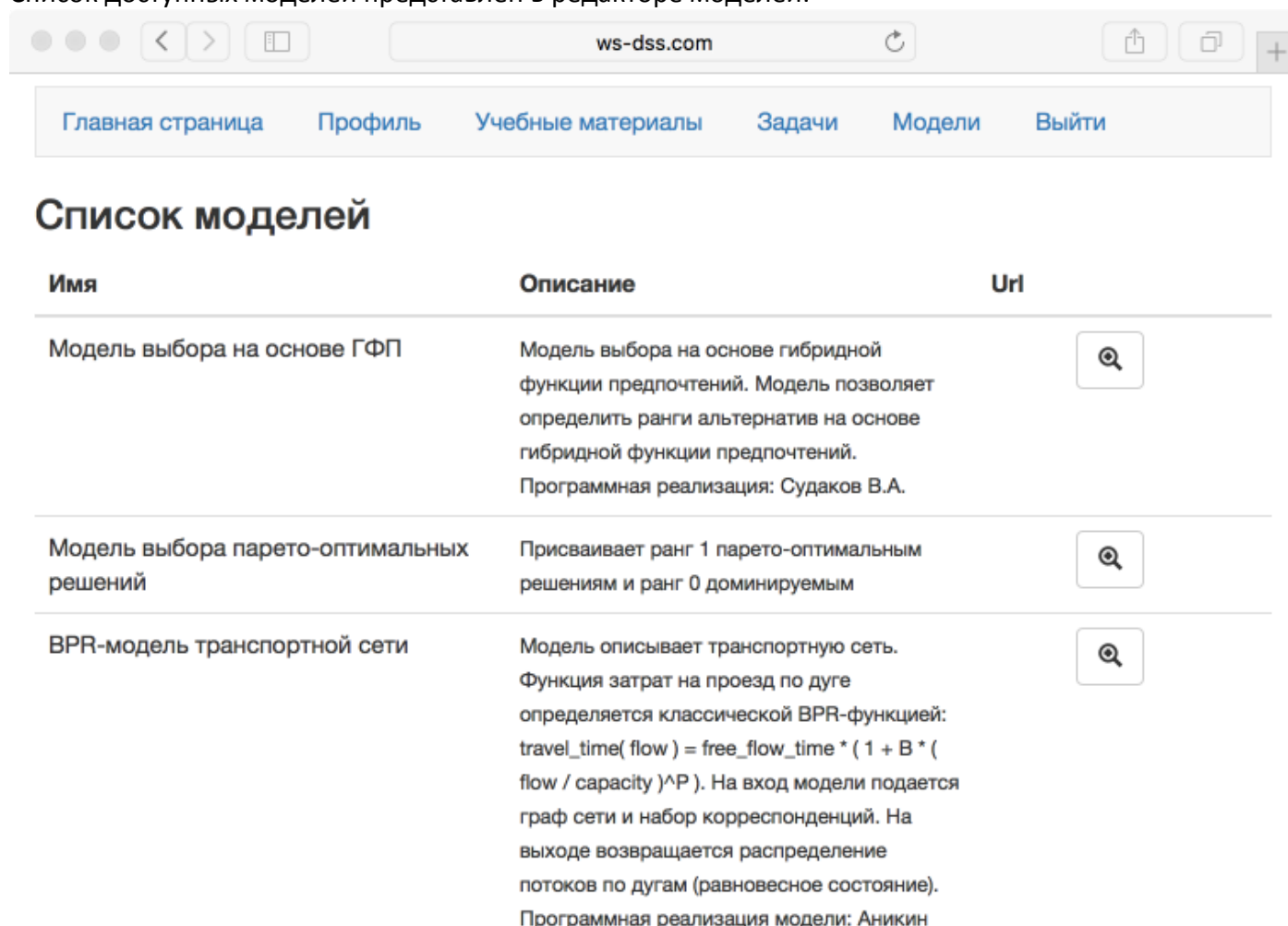
В основном меню появится пункт модели:






The screenshot shows a web browser window with the URL ws-dss.com. The main menu includes links for "Главная страница", "Профиль", "Учебные материалы", "Задачи", "Модели", and "Выйти". Below the menu, there is a section titled "Интеграция моделей" with sub-links: "Редактор моделей", "Прогоны моделей", "Множества прогонов моделей", and "Источники значений для прогонов".

© 2015 Владимир Судаков, Версия 1.1

Список доступных моделей представлен в редакторе моделей:



The screenshot shows the "Список моделей" (List of models) page. It features a table with three columns: "Имя" (Name), "Описание" (Description), and "Url". Each row includes a magnifying glass icon in the "Url" column.

Имя	Описание	Url
Модель выбора на основе ГФП	Модель выбора на основе гибридной функции предпочтений. Модель позволяет определить ранги альтернатив на основе гибридной функции предпочтений. Программная реализация: Судаков В.А.	
Модель выбора парето-оптимальных решений	Присваивает ранг 1 парето-оптимальным решениям и ранг 0 доминируемым	
BPR-модель транспортной сети	Модель описывает транспортную сеть. Функция затрат на проезд по дуге определяется классической BPR-функцией: $travel_time(flow) = free_flow_time * (1 + B * (flow / capacity)^P)$. На вход модели подается граф сети и набор корреспонденций. На выходе возвращается распределение потоков по дугам (равновесное состояние). Программная реализация модели: Аникин	

По-умолчанию обычный пользователь может только использовать созданные другими разработчиками модели. Если Вы хотите создавать и редактировать новые модели, то обратитесь за правами доступа и инструкцией к администратору системы.

Каждая модель характеризуется неким набором параметров:

Название	Описание	Целочисл.	Число измер.	Мин.знач.	Макс.знач.	Обязательный	Копирование
criteria_weight	Веса критериев	Нет	1	0.0		Да	Нет
criteria_dir	Направление улучшения критерия. Возможные значения: "max", "min"	Нет	1			Да	Нет
criteria_values	Значения критериев. Строка - критерий, Столбец - альтернатива.	Нет	2			Да	Нет
alternative_rank	Ранги альтернатив.	Нет	1	0.0	1.0	Нет	Нет

Входные параметры, обычно отмечены как обязательные. Выходные параметры необязательны. Таким образом для представленной модели первые 3 параметра выходные, а четвертый параметр – выходной.

Перейдите в основное меню Модели. Выберите пункт прогоны моделей. Создайте новый прогон требуемой модели:

Новый прогон модели

Название

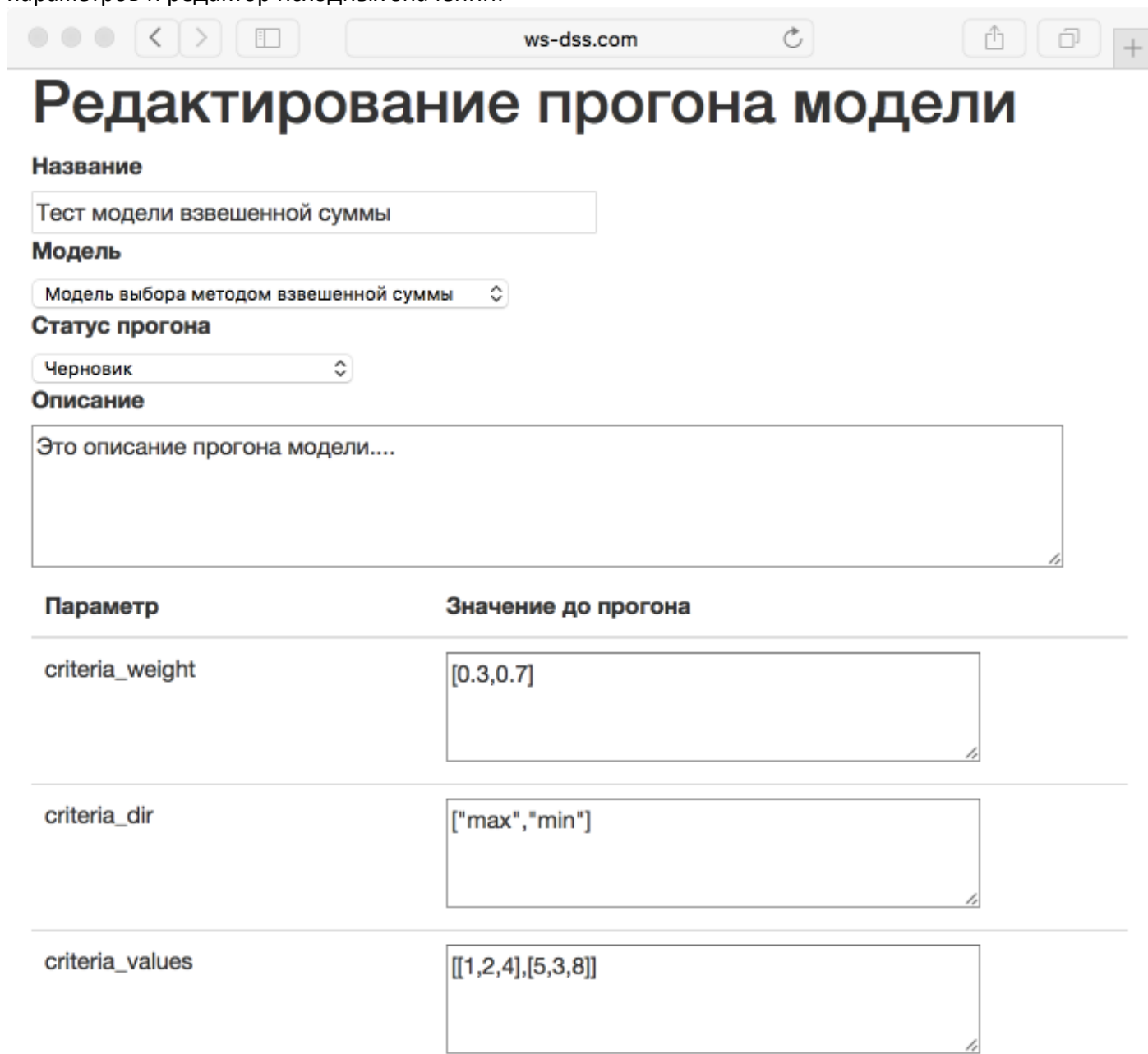
Модель

Статус прогона

Описание

Сохраните прогон модели.

После сохранения прогона вернуться в режим редактирования прогона - у Вас появится список параметров и редактор исходных значений:



Редактирование прогона модели

Название
Тест модели взвешенной суммы

Модель
Модель выбора методом взвешенной суммы

Статус прогона
Черновик

Описание
Это описание прогона модели....

Параметр	Значение до прогона
criteria_weight	[0.3,0.7]
criteria_dir	["max","min"]
criteria_values	[[1,2,4],[5,3,8]]

Вектора и матрицы следует задавать как массивы JSON. Для того, чтобы просигнализировать модели о необходимости расчета, следует сменить статус с «Черновик» на «Новая».

Сохраните прогон модели.

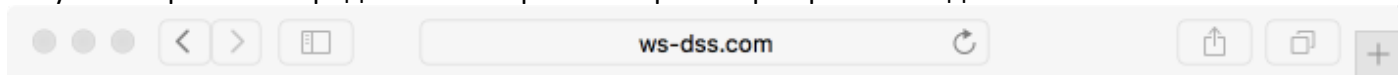
Когда модель начнет прогон, то она должна сменить статус «Новая» на статус «Идет прогон».

После завершения прогона модель должна установить статус в «Прогон завершен».

В случае если обнаружено, что комбинация значений параметров недопустима, то модель установит статус прогона в «Недопустимые параметры». Для прочих ошибок используется статус «Ошибка выполнения».

Как быстро будет получен результат зависит от специфики конкретной модели. Некоторые модели завершают расчеты за несколько секунд, а некоторые могут рассчитываться несколько часов.

Результаты расчетов представлены в режиме просмотра прогона модели:



Название: Тест модели взвешенной суммы

Модель: Модель выбора методом взвешенной суммы

Статус прогона: Прогон завершен

Описание:

Это описание прогона модели....

Трассировочная печать:

```
Число альтернатив: 3
Число критериев: 2
Нижние границы значений: [1, 3]
Верхние границы значений: [4, 8]
Нормированные значения весов: [0.3, 0.7]
Нормированные значения критериев: [[0.0, 0.3333, 1.0], [0.6, 1.0, 0.0]]
```

Доступность:

Значения параметров

Параметр	Значение до прогона	Значение после прогона
alternative_rank		[0.42, 0.8, 0.3]
criteria_values	[[1,2,4],[5,3,8]]	
criteria_dir	["max","min"]	
criteria_weight	[0.3,0.7]	

В данном случае получен массив рангов альтернатив (параметр **alternative_rank**).

Использование моделей из внешних приложений

Основу интеграции положена концепция репрезентативного состояния [https://ru.wikipedia.org/wiki/REST]. WS-DSS поддерживает некоторое количество ресурсов. Доступ к ресурсам организован по http. Методы доступа:

- GET – запросить информацию о ресурсах (аналог SELECT)
- POST – создать новый экземпляр ресурса (аналог INSERT)
- PUT – изменить экземпляр ресурса (аналог UPDATE)
- DELETE – удалить экземпляр ресурса (аналог DELETE)

Формат ответов сервера: JSON.

Тестирование взаимодействия с сервером можно проводить из командной строки с помощью стандартной программы **curl** [https://ru.wikipedia.org/wiki/CURL]. Однако curl – это только пример программы для *тестирования* программного взаимодействия. В большинстве современных языков/систем программирования встроены библиотеки выполнения http запросов. Именно такие библиотеки и следует использовать при выполнении http запросов.

Авторизация в ws-dss происходит по токenu «**user-token**» в *заголовке* http запроса. Для того чтобы узнать свой токен, нужно зайти на страницу «**Профиль**» и нажать кнопку «**Показать**» рядом с надписью «**Ваш токен**».

Пример простейшего запроса:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_models.json" -X GET
```

Ответ на него:

```
[{"id":10,"name":"Модель выбора на основе ГФП","descr":"Модель выбора на основе гибридной функции предпочтений. Модель позволяет определить ранги альтернатив на основе гибридной функции предпочтений. Программная реализация: Судаков В.А.\t", "model_url":null,"ws_method_id":2,"user_id":1,"ws_param_models":[{"id":51,"ws_model_id":10,"ws_param_id":42,"is_required":true,"created_at":"2016-05-19T14:03:15.348+03:00","updated_at":"2016-05-19T14:03:15.348+03:00","is_copy":false}, {"id":52,"ws_model_id":10,"ws_param_id":43,"is_required":true,"created_at":"2016-05-19T14:03:15.353+03:00","updated_at":"2016-05-19T14:03:15.353+03:00","is_copy":false}, {"id":53,"ws_model_id":10,"ws_param_id":14,"is_required":true,"created_at":"2016-05-19T14:03:15.355+03:00","updated_at":"2016-05-19T14:03:15.355+03:00","is_copy":false}, {"id":54,"ws_model_id":10,"ws_param_id":16,"is_required":true,"created_at":"2016-05-19T14:03:15.357+03:00","updated_at":"2016-05-19T14:03:15.357+03:00","is_copy":false}, {"id":55,"ws_model_id":10,"ws_param_id":17,"is_required":false,"created_at":"2016-05-19T14:03:15.359+03:00","updated_at":"2016-05-19T14:03:15.359+03:00","is_copy":false}, {"id":56,"ws_model_id":10,"ws_param_id":39,"is_required":true,"created_at":"2016-05-19T14:03:15.360+03:00","updated_at":"2016-05-19T14:03:15.360+03:00","is_copy":false}, {"id":57,"ws_model_id":10,"ws_param_id":40,"is_required":true,"created_at":"2016-05-19T14:03:15.362+03:00","updated_at":"2016-05-19T14:03:15.362+03:00","is_copy":false}, {"id":58,"ws_model_id":10,"ws_param_id":41,"is_required":true,"created_at":"2016-05-19T14:03:15.363+03:00","updated_at":"2016-05-19T14:03:15.363+03:00","is_copy":false}], "url":"http://ws-dss.com/ws_models/10.json"}, ... ]
```

Для фильтрации в запросах GET можно использовать предикаты <https://github.com/activerecord-hackery/ransack/wiki/basic-searching> в параметре q:

```
curl --header "user-token: ??????" http://ws-dss.com/ws_models.json" -X GET -d "q[id_eq]=6"
```

- этот пример фильтрует записи у которых id = 6.

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_models.json" -X GET -d "q[name_cont]=взвеш"
```

- этот пример возвращает записи у которых поле name содержит подстроку «взвеш» (name like '%взвеш%')

В строке приведённой строке URL `ws_models` – это имя ресурса. Для пользователя моделей могут понадобиться следующие ресурсы:

`ws_models` – модели

`ws_params` – параметры

`ws_model_runs` – прогоны

`ws_param_values` – значения параметров

`ws_model_statuses` – статусы прогона

Исходные тексты каркаса веб-сервиса находятся в открытом доступе <https://github.com/sudakov/ws-dss>. С ними знакомится не обязательно, но с академической точки зрения, может быть кому-то будет интересно.

Последовательность работы с сервисом:

Найти ид своей модели и ид своих параметров в результатах запроса:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_models.json" -X GET
```

В выше приведенном ответе "id":10 - это ид модели, "ws_param_id":43 – это ид одного из параметров.

Посмотреть имя параметра можно с помощью запроса:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_params.json" -X GET -d "q[id_eq]=43"
```

Или с помощью запроса:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_params/43.json" -X GET
```

Используемые Вами ИД моделей и ИД параметров проще всего прописать прямо в коде как именованные константы или хранить в настройках (ini файлы или настроечные таблицы в БД).

Посмотреть ид статусов прогона можно запросом:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_model_statuses.json" -X GET
[
{"id":1,"name":"Черновик","url":"http://ws-dss.com/ws_model_statuses/1.json"},
{"id":2,"name":"Новая","url":"http://ws-dss.com/ws_model_statuses/2.json"},
{"id":3,"name":"Идет прогон","url":"http://ws-dss.com/ws_model_statuses/3.json"},
{"id":4,"name":"Прогон завершен","url":"http://ws-dss.com/ws_model_statuses/4.json"},
{"id":5,"name":"Недопустимые параметры","url":"http://ws-dss.com/ws_model_statuses/5.json"},
{"id":6,"name":"Ошибка выполнения","url":"http://ws-dss.com/ws_model_statuses/6.json"},
{"id":7,"name":"Подготовка исходных данных","url":"http://ws-dss.com/ws_model_statuses/7.json"}]
```

Работа с требуемой моделью состоит из следующих шагов:

1. Создать прогон с требуемым ид модели в статусе «Черновик»:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_model_runs.json" -X POST -d
"ws_model_run[ws_model_id]=6&ws_model_run[name]=Тест 123&ws_model_run[ws_model_status_id]=1"
```

В ответе будет ид созданного прогона:

```
{"id":77, "name":"Тест 1234","ws_model_id":6,"ws_model_status_id":1,
"trace":null,"descr":null,"created_at":"2016-09-27T11:39:08.550+03:00",
"updated_at":"2016-09-27T11:39:08.550+03:00","ws_set_model_run_id":null,
"target_ws_model_id":null,"goal_ws_param_value_id":null,"ws_param_values":[]}
```

2. Создать записи `ws_param_values`, которые нужно считать:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_param_values.json" -X POST -d
"ws_param_value[ws_model_run_id]=77&ws_param_value[ws_param_id]=14&ws_param_value[old_value]=[0.1,0.9]"
```

В ответе содержится `id` записи, которую можно модифицировать:

```
{"id":362,"ws_param_id":14,"ws_model_run_id":77,"old_value":"1","new_value":null,"formula":null,"created_at":
"2016-09-27T11:46:33.962+03:00","updated_at":"2016-09-27T11:46:33.962+03:00"}
```

Например, для взвешенной суммы нужно создать 3 записи со значениями параметров.

Изменить статус прогона на «Новая»:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_model_runs/77.json" -X PUT -d
"ws_model_run[ws_model_status_id]=2"
```

Здесь вместо 77 подставить ид Вашего прогона.

Поскольку модель работает асинхронно, то нужно периодически опрашивать прогон, пока не будет получен один из финальных статусов: «Прогон завершен», «Недопустимые параметры», «Ошибка выполнения»:

```
curl --header "user-token: ??????" --url "http://ws-dss.com/ws_model_runs/77.json" -X GET
```

Ответ содержит не только статус, но и посчитанные выходные значения параметров:

```
{"id":77,"name":"Тест 1234","ws_model_id":6,"ws_model_status_id":4,"trace":"Число альтернатив: 3\nЧисло критериев: 2\nНижние границы значений: [1, 3]\nВерхние границы значений: [4, 8]\nНормированные значения весов:[0.1, 0.9]\nНормированные значения критериев:[[0.0, 0.3333, 1.0], [0.6, 1.0, 0.0]]",
"descr":null,"created_at":"2016-09-27T11:39:08.550+03:00","updated_at":"2016-09-27T16:44:09.057+03:00","ws_set_model_run_id":null,"target_ws_model_id":null,"goal_ws_param_value_id":null,
"ws_param_values": [
{"id":366,"ws_param_id":17,"ws_model_run_id":77,"old_value":null,
"new_value":"[0.54, 0.933, 0.1]",
"created_at":"2016-09-27T16:44:09.051+03:00","updated_at":"2016-09-27T16:44:09.051+03:00",
"ws_set_model_run_id":null,"formula":null},
{"id":365,"ws_param_id":15,"ws_model_run_id":77,"old_value":"[\"max\", \"min\"]","new_value":null,
"created_at":"2016-09-27T11:54:47.522+03:00", "updated_at":"2016-09-27T11:54:47.522+03:00",
"ws_set_model_run_id":null,"formula":null},
{"id":364,"ws_param_id":16,"ws_model_run_id":77,"old_value":"[[1,2,4], [5,3,8]]","new_value":null,
"created_at":"2016-09-27T11:53:43.774+03:00", "updated_at":"2016-09-27T11:53:43.774+03:00",
"ws_set_model_run_id":null,"formula":null},
{"id":363,"ws_param_id":14,"ws_model_run_id":77,"old_value":"[0.1,0.9]","new_value":null,
"created_at":"2016-09-27T11:50:48.347+03:00","updated_at":"2016-09-27T11:50:48.347+03:00",
"ws_set_model_run_id":null,"formula":null}]}
```

В данном случае нас интересовал параметр с ид = 17. Выходное значение записывается в поле `new_value`.

Работа с задачами из внешних приложений

Кроме моделей, ws-dss содержит более простой механизм задач. Он проще так как не требуется передавать параметры отдельными запросами, а исходные данные передаются сразу при создании задачи. К недостаткам относится то, что данный механизм не позволяет запускать модели, которые расположены на удаленных серверах. Кроме того отсутствует возможность задавать связи между отдельными подмоделями.

Список доступных через механизм задач методов можно получить запросом:

```
curl --header "user-token: ????" --url "http://ws-dss.com/ws_methods.json" -X GET
```

Здесь можно узнать требуемый ИД метода.

Все входные данные в задачу передаются как единый json в параметре input. Пример запрос на создание задачи:

```
curl --header "user-token: ????" --url "http://ws-dss.com/ws_jobs.json" -X POST -d  
"ws_job[ws_method_id]=8&ws_job[input]={\"team\": 20150310700}"
```

Ответ содержит ид задачи:

```
{\"id\":874,\"created_at\":\"2016-09-27T17:05:42.702+03:00\",\"updated_at\":\"2016-09-27T17:05:42.725+03:00\",  
\"input\":{\"team\": 20150310700},\"output\":\"\", \"ws_method_id\":8}
```

Ответ записывается в поле output. Задача обрабатывается асинхронно поэтому в поле output может сразу быть результата. В этом случае нужно периодически опрашивать созданную задачу:

```
curl --header "user-token: ????" --url "http://ws-dss.com/ws_jobs/874.json" -X GET
```

Обработанный ответ будет содержать непустое поле output:

```
{\"id\":874,\"created_at\":\"2016-09-27T17:05:42.702+03:00\",\"updated_at\":\"2016-09-27T17:05:42.725+03:00\",  
\"input\":{\"team\": 20150310700},  
\"output\":\"Даны ФАЛ, заданные следующими изображающими числами:\n1) 00100010010001001101110101110111\n00100010111011100111011101010101\nУказать сущетвенные и фиктивные агрументы\n\",  
\"ws_method_id\":8}
```