
Read the Docs Template Documentation

Выпуск 0.1-alpha

Read the Docs

апр. 13, 2020

1	Введение	1
1.1	Краткое описание новых возможностей и улучшений	1
2	Обновление функций версии hamming	3
2.1	Новые параметры для импорта из LDAP	3
2.2	Индексация гео-координат	3
2.3	Новая реализация push-уведомлений в мобильном приложении	4
2.4	Реализация фильтрации по статусу маршрута записи реестра	5
2.5	Методы API и события	5
2.5.1	Использование сессии Synergy в REST API	5
2.5.2	Изменение логина и пароля пользователя	7
2.5.3	Использование кодов для групп пользователей	7
2.5.4	Повторная отправка блокирующего процесса в очередь	8
2.5.5	Проверка принадлежности пользователя к подразделению	8
2.5.5.1	Получение всех подразделений пользователя	8
2.5.5.2	Проверка принадлежности пользователя определенному подразделению	10
2.5.6	Передача значений произвольных компонентов при поиске по реестру	10
2.5.7	События открытия и закрытия документов	10
2.6	Миграция данных в хранилище Cassandra	11
2.7	Настройка количества символов для поиска и сортировки текста	11
2.8	Увеличение допустимого размера скрипта в Пользовательском компоненте	11
2.9	Доработка экспорта данных в Excel	11
2.9.1	Экспорт переводимых компонентов	11
2.9.2	Экспорт даты	12

Мы постоянно разрабатываем новые возможности Платформы для того, чтобы упростить реализацию бизнес-приложений на ее основе, и приняли решение перенести некоторые из новых функций в версию `hamming`.

Поэтому версию Synergy 4.0-r1~190318.174652 мы рады объявить как **PowerPack2**. Она находится, как и обычно, в репозитории `hamming`.

1.1 Краткое описание новых возможностей и улучшений

- гибкая настройка расписания синхронизации с LDAP и AD, автоматическое добавление новых пользователей в указанную группу и другие новые настройки конфигурационного файла: *Новые параметры для импорта из LDAP*
- теперь можно визуализировать данные реестров на географической карте, используя координаты, благодаря доработке индексации компонентов форм: *Индексация гео-координат*
- новая реализация push-уведомлений в мобильном приложении поддерживает новейшие версии Android и iOS: *Новая реализация push-уведомлений в мобильном приложении*
- группируйте записи в фильтрах реестров не только по значениям их компонентов, но и по статусам маршрутов исполнения: *Реализация фильтрации по статусу маршрута записи реестра*
- основная часть новых возможностей - методы API и события Synergy:
 - решена проблема с коллизией авторизационных данных при работе одновременно с клиентской частью Synergy и внешними модулями: *Использование сессии Synergy в REST API*
 - если вы когда-либо реализовывали для своих клиентов порталное решение, то наверняка вам приходилось реализовывать и свои методы изменения логина и пароля для пользователей, а потом самим поддерживать эту реализацию. Новый метод API позволит изменить авторизационные данные пользователей портала средствами Synergy: *Изменение логина и пароля пользователя*

- для обращения к группам пользователей посредством API теперь достаточно только кодов групп, а не их внутренних UUID: *Использование кодов для групп пользователей*
- новый метод API позволяет перезапускать зависшие блокпроцессы в документе, вне зависимости от автора процесса: *Повторная отправка блокирующего процесса в очередь*
- реализованы методы, упрощающие реализацию логики, зависящей от принадлежности пользователя к подразделению оргструктуры, причем новые методы не зависят от UUID пользователей и объектов оргструктуры: *Проверка принадлежности пользователя к подразделению*
- новый параметр в API поиска по реестру позволяет одним запросом получить данные произвольных компонентов формы, а не только «отображаемых» полей реестра: *Передача значений произвольных компонентов при поиске по реестру*
- новые события открытия и закрытия документов в Synergy позволяют реализовать произвольную обработку этих действий пользователя - например, поставить защиту на одновременную работу нескольких пользователей с одним документом: *События открытия и закрытия документов*
- новый механизм миграции данных из хранилища Jackrabbit в Cassandra работает значительно быстрее, чем стандартный: *Миграция данных в хранилище Cassandra*
- новые параметры индексов позволяют более тонко настраивать количество символов, по которым выполняется поиск и сортировка документов, и за счет этого оптимизировать потребление ресурсов: *Настройка количества символов для поиска и сортировки текста*
- подключайте любые внешние библиотеки без необходимости ожидать, пока они загрузятся - объем данных в скриптах пользовательских компонентов теперь это позволяет: *Увеличение допустимого размера скрипта в Пользовательском компоненте*
- все переводимые данные форм при экспорте данных реестра в Excel теперь скачиваются в текущей локале, а не в локале последнего сохранения *Доработка экспорта данных в Excel*

Обновление функций версии hamming

2.1 Новые параметры для импорта из LDAP

Для упрощения процесса импорта данных из LDAP и ActiveDirectory мы реализовали новые параметры конфигурационного файла `/opt/synergy/jboss/standalone/configuration/arta/ldap-sync.xml`:

- **schedules** - расписание синхронизации (по времени сервера); данный параметр альтернативен интервалу синхронизации **interval**.

Параметр может содержать несколько таймеров, когда должна выполняться синхронизация. Если к моменту начала очередной синхронизации не была завершена предыдущая, то таймер пропускается.

- **defaultAccess** - предоставлять ли импортируемым пользователям доступ в систему, пока позволяет лицензия.
- **defaultGroup** - код группы пользователей, в которую нужно включить всех импортированных пользователей.
- **importGroups** - импортировать ли группы из LDAP/AD. Если в параметре указано значение **false**, то при импорте группы будут проигнорированы.

Полное описание параметров и пример их использования в конфигурационном файле приведено в Руководстве администратора.

2.2 Индексация гео-координат

Kibana поддерживает работу с картами, позволяя визуализировать данные, привязанные к координатам. В новой версии работу с координатами стала поддерживать и Synergy.

Гео-координаты индексируются в шаблоны индексов Kibana как поля с кодом `<код_компонента>_key_geo_point`. Это поле создается только в том случае, если из значения `key` компонента удалось выделить координаты, т.е. значение `key` соответствует формату «широта,

долгота» - содержит пару чисел, разделенных запятой и являющихся валидными координатами, например:

- 51.133333,71.433333 (пара чисел, разделенных запятой, без пробелов)
- 51.13333, 71.43333 (пара чисел, разделенных запятой и пробелом)
- 51.13, -71.43 (пара чисел с точностью до сотых, разделенные запятой и пробелом)
- 51.133, -71

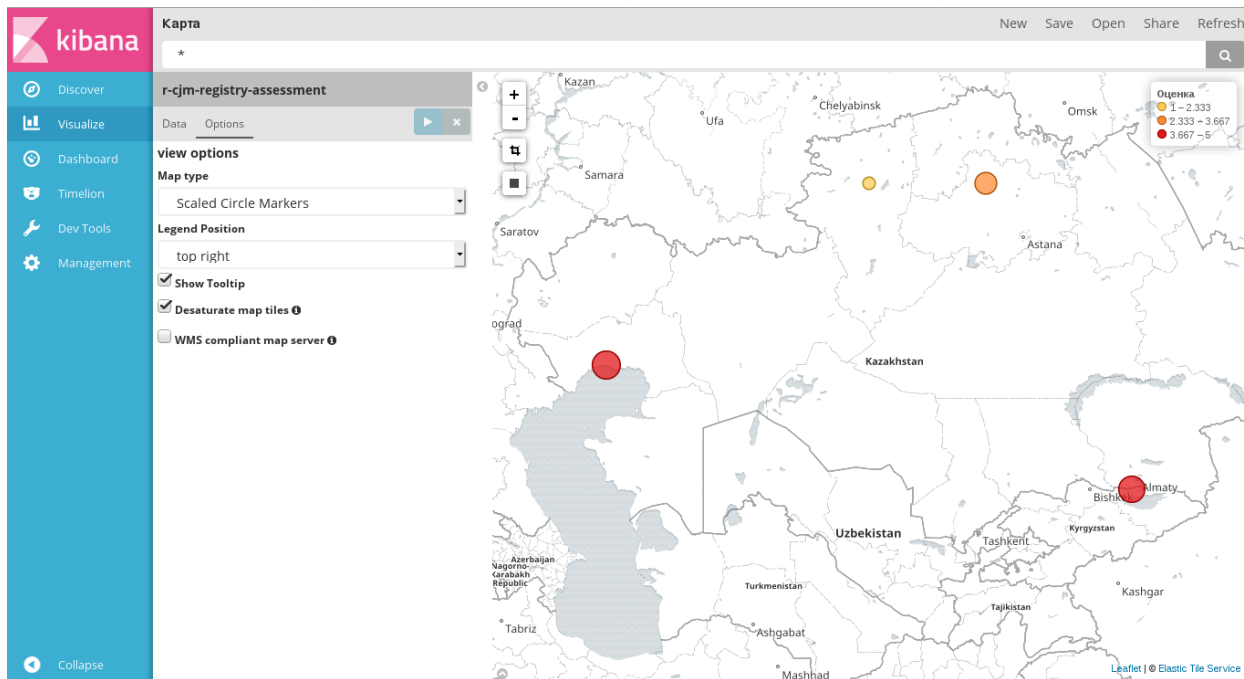


Рис. 1: Пример построения диаграммы по координатам

О том, как строить диаграмму карты в Kibana, рассказано в [Руководстве разработчика](#).

2.3 Новая реализация push-уведомлений в мобильном приложении

Ранее уведомления, формируемые на стороне сервера, не доходили на некоторые смартфоны на базе ОС Android и iOS. В первую очередь это касалось наиболее свежих версий Android 8 (Oreo) / 9 (Pie), iOS 11 / 12. Для решения проблемы была реализована работа с push-уведомлениями через Firebase Cloud Messaging.

Подробнее про технологию можно прочитать на [официальной странице Firebase](#).

Для корректной работы уведомлений требуется обновить как сервер Synergy, так и мобильные клиенты.

2.4 Реализация фильтрации по статусу маршрута записи реестра

Мы реализовали возможность фильтровать записи реестров не только по их содержимому, но и по статусу выполнения маршрутов этих записей.

Пример настройки условия по статусу маршрута:

The screenshot shows a web interface for managing filters. The main form has the following fields:

- Название** (Name): Черновики и незавершенные
- Код** (Code): chernoviki_i_nezavershennye
- Статус записи реестра** (Registry record status): A dropdown menu with 'Подготовка' and 'В процессе' selected.
- Права на реестр** (Registry rights): A section with a 'Группа' (Group) dropdown set to 'В' and a list of rights: 'Активная', 'Неуспешная', 'Изменение', and 'Удаление'.

A 'Добавить' (Add) button is located to the right of the 'Статус записи реестра' dropdown.

В одном условии на статус записи можно выбрать несколько значений.

Получившийся фильтр будет содержать только те записи, статус маршрута которых соответствует одному из выбранных в условии фильтра:

2.5 Методы API и события

Мы продолжаем развивать функциональность методов API.

2.5.1 Использование сессии Synergy в REST API

Synergy REST API использовал сессии для запросов, из-за чего при передаче в запросе cookie от Synergy для аутентификации использовались данные сессии, а не то, что было передано в Authorization.

Внешне эту проблему можно было наблюдать, если в одной вкладке браузера авторизоваться в клиенте /Synergy, а в соседней - во внешнем модуле под другим пользователем. В этом случае все запросы во внешнем модуле выполнялись от имени того пользователя, который авторизован в основном клиенте Synergy.

В новой версии hamming эта проблема была исправлена: теперь авторизация не смешивается, и в запросах API используются только переданные данные Authorization, а не данные сессии.

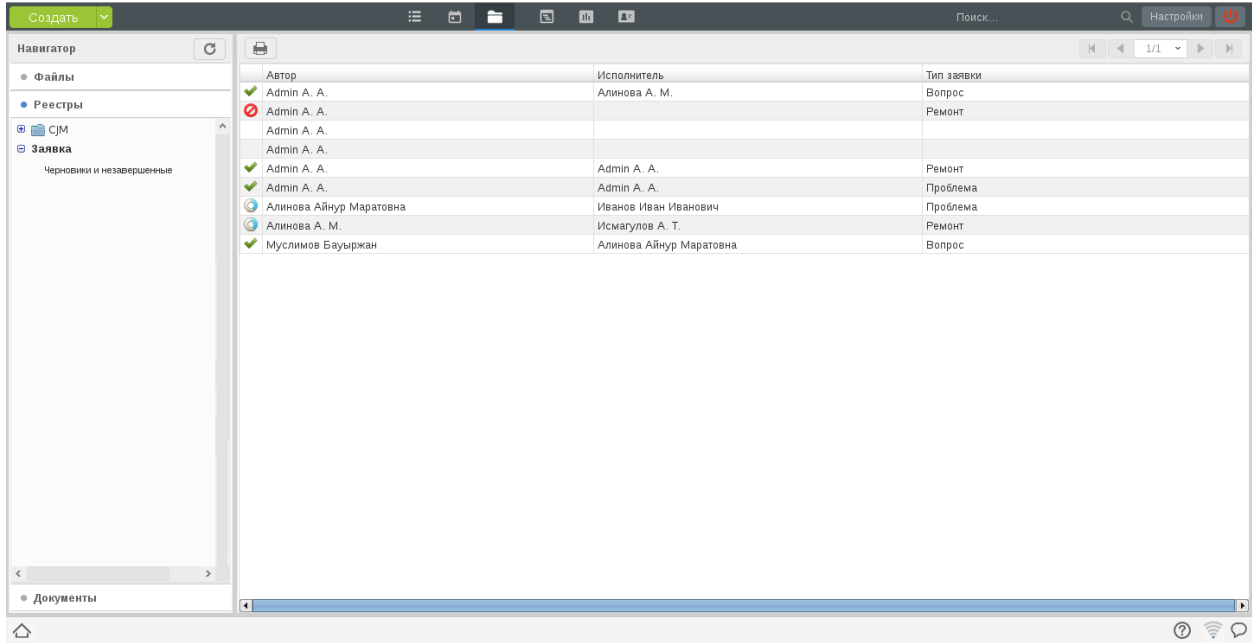


Рис. 2: Все записи реестра

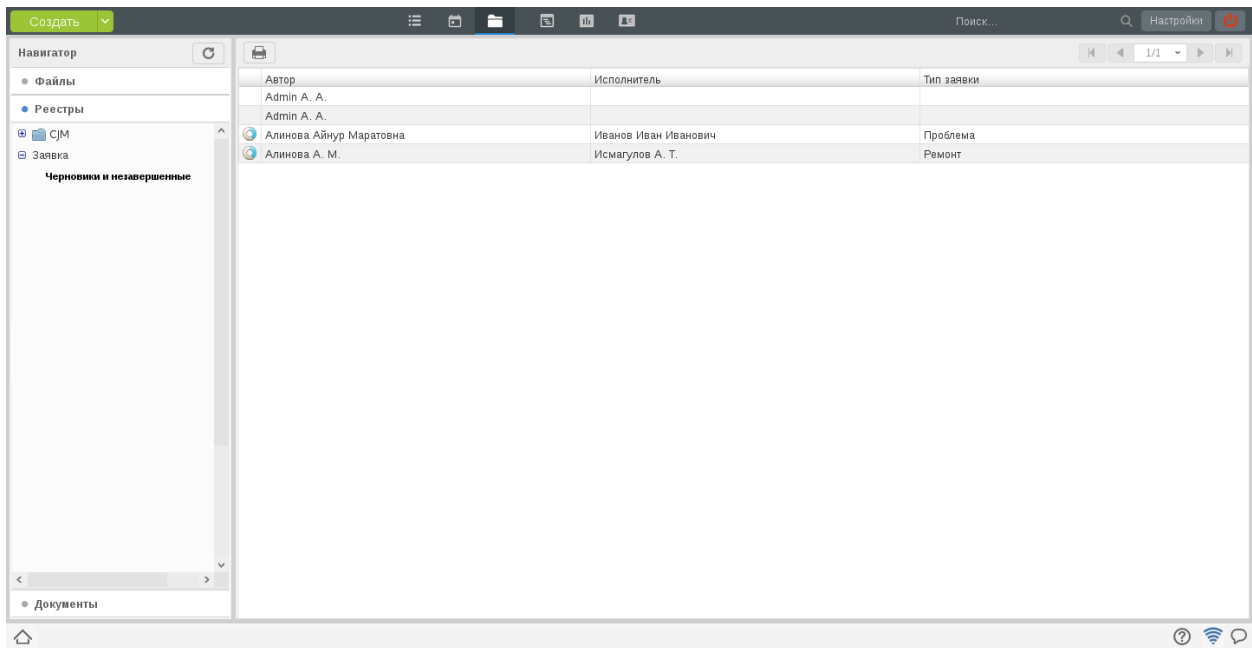


Рис. 3: Записи фильтра, статусы которых «Подготовка» и «В процессе»

2.5.2 Изменение логина и пароля пользователя

Мы реализовали новый метод, позволяющий изменить логин и/или пароль авторизованного пользователя. Рекомендуем использовать этот метод в порталных решениях на базе Synergy.

URL метода: `rest/api/filecabinet/user/changeCredentials`

Тип: `POST`

Параметры метода:

- `actionCode` - тип действия (обяз.):
 - `CHANGE_LOGIN` - изменить логин
 - `CHANGE_PASSWORD` - изменить пароль
 - `CHANGE_ALL` - изменить и логин, и пароль пользователя
- `login` - новый логин пользователя; обязателен, если параметр `actionCode` имеет значение `CHANGE_LOGIN` или `CHANGE_ALL`
- `password` - новый пароль пользователя; обязателен, если параметр `actionCode` имеет значение `CHANGE_PASSWORD` или `CHANGE_ALL`
- `passwordConfirm` - подтверждение пароля, значение параметра должно совпадать с параметром `password`; обязателен, если параметр `actionCode` имеет значение `CHANGE_PASSWORD` или `CHANGE_ALL`
- `locale` - локаль пользователя (не обязателен, по умолчанию используется `ru`).

Полное описание метода приведено в `swagger`.

2.5.3 Использование кодов для групп пользователей

Исторически сложилось, что во всех методах API, оперирующих группами пользователей, невозможно было использовать код этой группы: методы принимали и возвращали только `UUID` объекта группы.

Мы доработали все методы, работающие с группами пользователей, добавив в их принимаемые и/или возвращаемые параметры новый параметр `groupCode`. Перечень измененных методов:

- `rest/api/storage/groups/list`
- `rest/api/userchooser/search`
- `rest/api/userchooser/search_ext`
- `rest/api/userchooser/getUserInfo`
- `rest/api/storage/groups/add_user`
- `rest/api/storage/groups/remove_user`
- `rest/api/storage/groups/list`
- `rest/api/groups/content`
- `rest/api/groups/find`
- `rest/api/person/auth`

2.5.4 Повторная отправка блокирующего процесса в очередь

Теперь решать проблемы с «зависшими» блокирующими процессами в маршрутах документов стало проще. Мы реализовали новый метод API, повторно отправляющий все зависшие блокирующие процессы в указанном документе в очередь.

URL метода: `rest/api/processes/retry_bp`

Тип: `POST`

Параметр:

- `documentID` - идентификатор документа, в котором требуется переотправить БП.

Примечание: Метод должен выполняться от имени пользователя с ролью «Разработчик Synergy».

Полное описание метода приведено в [swagger](#).

2.5.5 Проверка принадлежности пользователя к подразделению

Мы реализовали два новых метода, предназначенных для проверки принадлежности пользователя определенному родительскому подразделению. Причем для проверки могут использоваться как `UUID` подразделений и пользователей, так и их коды для показателей, что упрощает процесс переноса приложений.

2.5.5.1 Получение всех подразделений пользователя

Метод возвращает список из `ID` и кодов подразделений, к которым принадлежит этот пользователь, начиная с его непосредственного подразделения и вверх до корня.

Описание метода приведено в [swagger](#).

URL метода: `rest/api/person/get_user_departments`

Тип: `GET`

Параметры:

- `userCode` - код для показателей пользователя
- `userID` - `UUID` пользователя (не обязателен, если передан `userCode`)

Если переданы одновременно и код, и `UUID` пользователя, будет использован только код.

Пример:

На сервере настроена оргструктура вида:

- `ROOT`
 - `Basic 1`
 - * `Линия 1`
 - должность «`Service manager`»
 - должность «`Test`»

Пользователь «Тестовый Сотрудник» назначен на должности «`Service manager`» и «`Test`».

Для этого пользователя метод `rest/api/person/get_user_departments` вернет массив `json` вида:

```
[
  [
    {
      "id": "1",
      "code": "root"
    }
  ],
  [
    {
      "id": "6fd171e5-dc6f-45da-a677-3d0dd75e4802",
      "code": "liniya_1"
    },
    {
      "id": "6262e38d-bcac-4ef6-acae-dfaf193c0cdc",
      "code": "basic"
    },
    {
      "id": "1",
      "code": "root"
    }
  ]
]
```

Метод вернул массив с двумя элементами:

1. элемент с корневой нодой структуры: признак того, что пользователь непосредственно принадлежит этой ноде:

```
[
  {
    "id": "1",
    "code": "root"
  }
]
```

2. элемент со списком из трех нод: «liniya_1», «basic» и «root»: признак того, что пользователь непосредственно входит в подразделение «Линия 1», далее перечислены родительские подразделения вплоть до корневой ноды «root»:

```
[
  {
    "id": "6fd171e5-dc6f-45da-a677-3d0dd75e4802",
    "code": "liniya_1"
  },
  {
    "id": "6262e38d-bcac-4ef6-acae-dfaf193c0cdc",
    "code": "basic"
  },
  {
    "id": "1",
    "code": "root"
  }
]
```

2.5.5.2 Проверка принадлежности пользователя определенному подразделению

Метод определяет, принадлежит ли указанный пользователь указанному подразделению или любому его из родительских подразделений вплоть до корня.

Описание метода приведено в *swagger*.

URL метода: `rest/api/person/check_user_department`

Тип: `GET`

Параметры:

- `depCode` - код для показателей подразделения
- `depID` - UUID подразделения (не обязателен, если передан `depCode`)
- `userCode` - код для показателей пользователя
- `userID` - UUID пользователя (не обязателен, если передан `userCode`)

Если переданы одновременно и код, и UUID пользователя/подразделения, будет использован только соответствующий код.

Метод возвращает `true`, если пользователь входит в указанное подразделение, и `false` в противном случае.

2.5.6 Передача значений произвольных компонентов при поиске по реестру

При разработке бизнес-приложений на Synergy часто возникает необходимость получить максимум данных из реестра одним запросом, а потом использовать полученные данные или фильтровать их на стороне клиента.

Такой вариант использования требует указания всех данных, которые необходимо получить из реестра, как отображаемых полей реестра. При этом получаемые данные могут иметь специфическую структуру - например, это может быть изображение, закодированное в `Base64`, или UUID данных.

Мы доработали метод поиска по реестру `rest/api/registry/data_ext`, добавив для него новый параметр `fields`. В этом параметре можно передать коды компонентов формы, которые нужно возвращать для каждой записи реестра.

Пример использования: `fields=text1&fields=text2&fields=number1`

Теперь один запрос данных реестра позволяет получить не только данные отображаемых полей реестра, но и данные произвольных компонентов формы этого реестра.

Полное описание метода приведено в *swagger*.

2.5.7 События открытия и закрытия документов

Реализованы новые события открытия и закрытия документов. Теперь можно реализовать произвольную обработку этих событий: например, запретить открывать документ, пока он открыт другим пользователем, чтобы избежать перезаписи данных.

Описания событий:

- `event.docflow.document.opened` - открытие документа (генерируется только из UI Synergy):
 - `documentID` - идентификатор документа;
 - `userID` - идентификатор пользователя, сохранившего данные по форме;

- `date` - дата и время открытия документа;
- `event.docflow.document.closed` - закрытие документа (генерируется только из UI Synergy):
 - `documentID` - идентификатор документа;
 - `userID` - идентификатор пользователя, сохранившего данные по форме;
 - `date` - дата и время открытия документа.

2.6 Миграция данных в хранилище Cassandra

Если хранилище Cassandra было подключено на замену стандартному хранилищу Jackrabbit, уже содержащему данные, необходимо выполнить предварительную миграцию данных.

Однако, если хранилище содержит большой объем данных, стандартная миграция будет выполняться долгое время (несколько суток). Для ускорения этой процедуры предлагаем использовать **кастомную миграцию**, которая предназначена для переноса только последних версий неудаленных документов реестров и личных карточек пользователей.

Процедура выполнения кастомной миграции приведена в [Руководстве администратора](#).

2.7 Настройка количества символов для поиска и сортировки текста

Новые настройки индексов Lucene и Elasticsearch позволяют более тонко регулировать количество символов, используемых для поиска и сортировки данных форм, реестров, документов и файлов.

Подсказка: Эти настройки полезны также при визуализации данных, поскольку регулируют количество символов, попадающих в поля индекса `_sort` и `_exact`. По умолчанию в этих полях используется 50 и 300 символов соответственно, из-за чего часть данных в полях обрезается.

Описание этих настроек приведено в [Руководстве администратора](#).

2.8 Увеличение допустимого размера скрипта в Пользовательском компоненте

Ранее при прямой вставке кода внешних библиотек в скрипт пользовательского компонента возникала ошибка сохранения компонента из-за слишком большого объема скрипта.

Мы увеличили доступный объем данных скрипта и `html` пользовательского компонента до 4Gb.

2.9 Доработка экспорта данных в Excel

2.9.1 Экспорт переводимых компонентов

Ранее при экспорте данных в Excel данных реестра либо фильтра реестра в столбцы, соответствующие компонентам формы с источником данных - справочником (т.е. «Выпадающий список», «Выбор ва-

риантов», «Переключатель вариантов») подставлялось значение локали, используемой пользователем, последним сохранившим данные формы. То же самое справедливо для компонента «Дата/время».

В результате результирующий файл Excel мог содержать значения переводимых элементов на разных языках.

В новой версии `hamming` мы исправили это поведение: теперь при экспорте данных в Excel в значения столбцов, которые могут иметь разное значение в зависимости от локали, подставлялись данные, соответствующие текущей локали сеанса пользователя Synergy.

2.9.2 Экспорт даты

Ранее при экспорте поля, содержащего дату, соответствующему столбцу в Excel присваивался тип «Числовой», что затрудняло дальнейшую работу с этими данными. Теперь при экспорте даты столбцы имеют тип «Дата».

Примечание:

Теперь при экспорте данных не сохраняется формат даты, заданный в компоненте формы: например, компонент «Дата» со значением "22" марта 2019г. будет экспортирован только как дата 22.03.19, без формата.
