

API Яндекс.Спеллера

Руководство разработчика

21.02.2012

Яндекс

API Яндекс.Спеллера. Руководство разработчика. Версия 1.0

Дата сборки документа: 21.02.2012.

Этот документ является составной частью технической документации Яндекса.

Сайт справки к сервисам Яндекса: <http://help.yandex.ru>

© 2008—2012 ООО «ЯНДЕКС». Все права защищены.

СБ ОРФО © ООО Информатик, 1995-2007. Все права защищены.

Предупреждение об исключительных правах

Яндексу (а также указанному им правообладателю) принадлежат исключительные права на все результаты интеллектуальной деятельности и приравненные к ним средства индивидуализации, используемые при разработке, поддержке и эксплуатации сервиса API Яндекс.Спеллера. К таким результатам могут относиться, но не ограничиваясь указанными, программы для ЭВМ, базы данных, изображения, тексты, другие произведения, а также изобретения, полезные модели, товарные знаки, знаки обслуживания, коммерческие обозначения и фирменные наименования. Эти права охраняются в соответствии с Гражданским кодексом РФ и международным правом.

Вы можете использовать сервис API Яндекс.Спеллера или его составные части только в рамках полномочий, предоставленных вам Пользовательским соглашением сервиса API Яндекс.Спеллера или специального соглашения.

Нарушение требований по защите исключительных прав правообладателя влечет за собой дисциплинарную, гражданско-правовую, административную или уголовную ответственность в соответствии с российским законодательством.

Контактная информация

ООО «ЯНДЕКС»

<http://www.yandex.ru>

Тел.: +7 495 739 7000

Email: pr@yandex-team.ru

Главный офис: 119021, Россия, г. Москва, ул. Льва Толстого, д. 16

Содержание

О руководстве	4
О сервисе	4
Web Service API	5
Метод checkText	5
Метод checkTexts	8
Настройки Яндекс.Спеллера	10
Коды ошибок	11
JavaScript API	12
Класс Speller	12
Подключение к веб-формам	14
Подключение к текстовым редакторам	16
Подключение к TinyMCE	16
Подключение к Drupal	17
Подключение к CKEditor	19
Подключение к Joomla!	20
Подключение к WordPress	20
Подключение к FCKEditor 2.6	21
Яндекс.Спеллер как замена Aspell/IsPELL	22
Интересные решения	23
Предметный указатель	24

О руководстве

Руководство содержит описание API сервиса проверки орфографии Яндекс.Спеллера, а также информацию о том, как подключить и использовать сервис.

Руководство предназначено для веб-разработчиков, которые хотят добавить на страницы своих сайтов возможность проверки орфографии.

Информация, представленная в руководстве, актуальна на апрель 2011 года.

О сервисе

Сервис проверки правописания **Яндекс.Спеллер** (коротко Спеллер) предлагает веб-разработчикам использовать возможность интерактивной проверки орфографии на страницах своих сайтов.

Спеллер позволяет выявлять и исправлять орфографические ошибки в текстах, набираемых посетителями сайтов в различных HTML-формах (например, в формах заказов для совершения покупок онлайн, анкетах, резюме, формах для отправки сообщений в блоги и т. п.).

Спеллер анализирует слова, основываясь на правилах орфографии и лексике современного языка. В качестве словарного источника используется орфографический словарь, содержащий правильные написания большинства наиболее употребимых слов.

Сервис в настоящее время поддерживает три языка:

- русский – словарь содержит 3.6 миллиона словоформ (разработка компании «Информатик»);
- украинский – словарь содержит 1.8 миллиона словоформ (разработка компании «Информатик»);
- английский – словарь содержит 150000 словоформ (разработка компании «Яндекс»).

Примечание:

Яндекс.Спеллер выявляет и исправляет только орфографические ошибки, то есть ошибки в отдельных словах, такие как:

- неправильные, пропущенные или лишние буквы, например, "карова", "велослед", "караввай";
- ошибки капитализации (неправильное употребление прописных и строчных букв), например, "мОСКВА";
- повторы слов, например, "поезд поехал *поехал* быстрее".

Пунктуационные, грамматические (ошибки согласования слов) и стилистические ошибки не исправляются.

Яндекс.Спеллер является HTTP-службой и предоставляет [Web Service API](#) для обработки клиентских запросов на проверку орфографии. Яндекс.Спеллер поддерживает HTTP GET- и POST-запросы и работает в кодировке UTF-8.

Совет:

Чтобы изменить кодировку входного запроса, следует передать дополнительный параметр `ie` (input encoding). Возможные значения: `ie=utf-8` или `ie=1251`.

Web Service API

Для доступа к Яндекс.Спеллеру по HTTP предлагаются XML-, SOAP-, JSON- и JSONP-интерфейсы. Все интерфейсы обеспечивают одинаковую функциональность и используют одни и те же входные параметры.

XML- и SOAP- интерфейсы возвращают ответ в виде XML-документа, JSON-интерфейс вместо XML-элементов возвращает JavaScript-объекты с теми же именами и семантикой, а JSONP-интерфейс возвращает те же самые JavaScript-объекты, но в виде вызова callback-функции с заданным именем.

Ограничение:

Ограничения на использование сервиса приведены в [Пользовательском соглашении](#).

Доступ к API предоставляется по следующим URL:

- XML- и SOAP-интерфейс:

```
http://speller.yandex.net/services/spellservice
```

Примечание:

WSDL-документ для SOAP доступен по адресу: <http://speller.yandex.net/services/spellservice?WSDL>.

- JSON- и JSONP-интерфейс:

```
http://speller.yandex.net/services/spellservice.json
```

Примечание:

Для включения JSONP-интерфейса требуется передать дополнительный параметр callback с именем функции обратного вызова.

Web Service API включает в себя два метода, которые позволяют проверять правописание в одном или нескольких фрагментах текстов:

- [Метод checkText](#)
- [Метод checkTexts](#)

Метод checkText

Описание

```
SpellResult checkText(string text, string lang, int options)
```

Проверяет орфографию в указанном отрывке текста.

Страница для демонстрации работы метода: <http://speller.yandex.net/services/spellservice?op=checkText>

Входные параметры

Входные параметры могут передаваться либо с помощью запроса HTTP GET (см. пример ниже), либо с помощью HTTP POST. Во втором случае параметры передаются в теле запроса.

См. также [замечание](#) о кодировке входных запросов.

Пример запроса:

XML-интерфейс:

```
http://speller.yandex.net/services/spellservice/checkText?text=синхрѡфазатрон+в+дубне
```

JSON-интерфейс:

```
http://speller.yandex.net/services/spellservice.json/checkText?text=синхрѡфазатрон+в+дубне
```

JSONP-интерфейс (для функции myCallback):

```
http://speller.yandex.net/services/spellservice.json/checkText?text=синхрѡфазатрон+в+дубне&callback=myCallback
```

Ниже приводится полный список входных параметров.

Параметр	Тип	Описание
Обязательные		
text	string	Текст для проверки. Ограничения: <ul style="list-style-type: none"> Для POST-запросов максимальный размер передаваемого текста составляет 10000 символов. В GET-запросах ограничивается не размер передаваемого текста, а размер всей строки запроса, которая кроме текста может содержать и другие параметры. Максимальный размер строки запроса — 10Кб (в Internet Explorer 6 и 7 — 2Кб).
Необязательные		
lang	string	Языки проверки (перечисляются через запятую). Возможные значения: <ul style="list-style-type: none"> ru – русский; uk – украинский; en – английский. По умолчанию: "ru, en".
options	int	Опции Яндекс.Спеллера. Значением параметра является сумма значений требуемых опций, см. Настройки Яндекс.Спеллера . Например, options=5 — это сумма опций IGNORE_UPPERCASE и IGNORE_URLS. По умолчанию options = 0.
format	string	Формат проверяемого текста. Возможные значения: <ul style="list-style-type: none"> plain — текст без разметки (значение по умолчанию); html — HTML-текст.
callback	string	Имя функции обратного вызова (только для JSONP-интерфейса). Например: "myCallback".

Возвращает

В XML-интерфейсе возвращает ответ в виде XML-документа с корневым элементом SpellResult.
Например:

```
<?xml version="1.0" encoding="utf-8" ?>
<SpellResult>
  <error code="1" pos="0" row="0" col="0" len="14">
    <word>синхрофазатрон</word>
    <s>синхрофазотрон</s>
  </error>
  <error code="3" pos="17" row="0" col="17" len="5">
    <word>дубне</word>
    <s>Дубне</s>
  </error>
</SpellResult>
```

Элементы XML-схемы ответа:

- SpellResult – корневой элемент;
- error – информация об ошибке (может быть несколько или могут отсутствовать);
- word – исходное слово;
- s – подсказка (может быть несколько или могут отсутствовать).

Элемент <error> содержит следующие атрибуты:

- code – код ошибки, см. [Коды ошибок](#);
- pos – позиция слова с ошибкой (отсчет от 0);
- row – номер строки (отсчет от 0);
- col – номер столбца (отсчет от 0);
- len – длина слова с ошибкой.

В JSON-интерфейсе вместо XML-элементов возвращаются JavaScript-объекты с теми же именами и семантикой:

```
[
  { "code": 1, "pos": 0, "row": 0, "col": 0, "len": 14,
    "word": "синхрофазатрон",
    "s": [ "синхрофазотрон" ]
  },
  { "code": 3, "pos": 17, "row": 0, "col": 17, "len": 5,
    "word": "дубне",
    "s": [ "Дубне" ]
  }
]
```

В JSONP-интерфейсе те же самые JavaScript-объекты возвращаются в callback-функции (например, myCallback):

```
myCallback([
  { "code": 1, "pos": 0, "row": 0, "col": 0, "len": 14,
    "word": "синхрофазатрон",
    "s": [ "синхрофазотрон" ]
  },
  { "code": 3, "pos": 17, "row": 0, "col": 17, "len": 5,
    "word": "дубне",
    "s": [ "Дубне" ]
  }
])
```

См. также
[Метод checkTexts](#)
[Коды ошибок](#)

Метод checkTexts

Описание

```
SpellResult[] checkTexts(string[] text, string lang, int options)
```

Проверяет орфографию в указанных фрагментах текста. Для каждого фрагмента возвращается отдельный массив ошибок с подсказками.

Страница для демонстрации работы метода: <http://speller.yandex.net/services/spellservice?op=checkTexts>

Входные параметры

Входные параметры могут передаваться либо с помощью HTTP GET-запроса (см. пример), либо с помощью HTTP POST-запроса, где параметры передаются в body HTTP-запроса.

См. также [замечание](#) о кодировке входных запросов.

Пример запроса:

XML-интерфейс:

```
http://speller.yandex.net/services/spellservice/checkTexts?
text=синхрафазатрон&text=в+дубне
```

JSON-интерфейс:

```
http://speller.yandex.net/services/spellservice.json/checkTexts?
text=синхрафазатрон&text=в+дубне
```

JSONP-интерфейс (для функции myCallback):

```
http://speller.yandex.net/services/spellservice.json/checkTexts?
text=синхрафазатрон&text=в+дубне&callback=myCallback
```

Ниже приводится полный список входных параметров.

Параметр	Тип	Описание
Обязательные		
text	string[]	Массив текстов для проверки. Ограничения: <ul style="list-style-type: none"> Для POST-запросов максимальный размер передаваемого текста составляет 10000 символов. В GET-запросах ограничивается не размер передаваемого текста, а размер всей строки запроса, которая кроме текста может содержать и другие параметры. Максимальный размер строки запроса — 10Кб (в Internet Explorer 6 и 7 — 2Кб).
Необязательные		
lang	string	Языки проверки (перечисляются через запятую). Возможные значения: <ul style="list-style-type: none"> ru – русский; uk – украинский; en – английский. По умолчанию: "ru,en".
options	int	Опции Яндекс.Спеллера. Значением параметра является сумма значений требуемых опций, см. таблицу ниже. Например, options=5 — это сумма опций IGNORE_UPPERCASE и IGNORE_URLS. По умолчанию options = 0.

Параметр	Тип	Описание
format	string	Формат проверяемого текста. Возможные значения: <ul style="list-style-type: none"> plain — текст без разметки (значение по умолчанию); html — HTML-текст.
callback	string	Имя функции обратного вызова (только для JSONP-интерфейса). Например: "myCallback".

Возвращает

В XML-интерфейсе возвращает элемент `ArrayOfSpellResult`, содержащий последовательность элементов `SpellResult` по одному для каждого фрагмента текста. Например:

```
<ArrayOfSpellResult>
  <SpellResult>
    <error code="1" pos="0" row="0" col="0" len="14">
      <word>синхрофазатрон</word>
      <s>синхрофазотрон</s>
    </error>
  </SpellResult>
  <SpellResult>
    <error code="3" pos="2" row="0" col="2" len="5">
      <word>дубне</word>
      <s>Дубне</s>
    </error>
  </SpellResult>
</ArrayOfSpellResult>
```

Элементы XML-схемы ответа:

- `ArrayOfSpellResult` – корневой элемент, содержит элементы `SpellResult` по одному для каждого фрагмента текста;
- `SpellResult` – результат проверки одного фрагмента текста. Если найдены ошибки, то элемент будет содержать последовательность элементов `error`.
- `error` – информация об ошибке (может быть несколько или могут отсутствовать);
- `word` – исходное слово;
- `s` – подсказка (может быть несколько или могут отсутствовать).

Элемент `<error>` содержит следующие атрибуты:

- `code` – код ошибки;
- `pos` – позиция слова с ошибкой (отсчет от 0);
- `row` – номер строки (отсчет от 0);
- `col` – номер столбца (отсчет от 0);
- `len` – длина слова с ошибкой.

В JSON-интерфейсе вместо XML-элементов возвращаются JavaScript-объекты с теми же именами и семантикой:

```
[
  [
    { "code": 1, "pos": 0, "row": 0, "col": 0, "len": 14,
      "word": "синхрафазатрон",
      "s": [ "синхрофазотрон" ]
    }
  ],
  [
    { "code": 3, "pos": 2, "row": 0, "col": 2, "len": 5,
      "word": "дубне",
      "s": [ "Дубне" ]
    }
  ]
]
```

В JSONP-интерфейсе те же самые JavaScript-объекты возвращаются в callback-функции (например, myCallback):

```
myCallback([
  [
    { "code": 1, "pos": 0, "row": 0, "col": 0, "len": 14,
      "word": "синхрафазатрон",
      "s": [ "синхрофазотрон" ]
    }
  ],
  [
    { "code": 3, "pos": 2, "row": 0, "col": 2, "len": 5,
      "word": "дубне",
      "s": [ "Дубне" ]
    }
  ]
])
```

См. также
[Метод checkText](#)
[Коды ошибок](#)

Настройки Яндекс.Спеллера

В таблице указан перечень настроек Яндекс.Спеллера.

Опция	Значение (десятичное)	Описание (пример)
IGNORE_UPPERCASE	1	Пропускать слова, написанные заглавными буквами, например, "ВПК".
IGNORE_DIGITS	2	Пропускать слова с цифрами, например, "авп17х4534".
IGNORE_URLS	4	Пропускать интернет-адреса, почтовые адреса и имена файлов.
FIND_REPEAT_WORDS	8	Подсвечивать повторы слов, идущие подряд. Например, "я полетел <i>на на</i> Кипр".
IGNORE_LATIN	16	Пропускать слова, написанные латиницей, например, "madrid".
NO_SUGGEST	32	Только проверять текст, не выдавая вариантов для замены.
FLAG_LATIN	128	Отмечать слова, написанные латиницей, как ошибочные.

Опция	Значение (десятичное)	Описание (пример)
BY_WORDS	256	Не использовать словарное окружение (контекст) при проверке. Опция полезна в случаях, когда на вход сервиса передается список отдельных слов.
IGNORE_CAPITALIZATION	512	Игнорировать неверное употребление ПРОПИСНЫХ/строчных букв, например, в слове "москва".

См. также

[Метод checkText](#)

[Метод checkTexts](#)

Коды ошибок

В таблице указан перечень кодов ошибок, возвращаемых методами API Яндекс.Спеллера.

Код	Ошибка	Краткое описание
1	ERROR_UNKNOWN_WORD	Слова нет в словаре.
2	ERROR_REPEAT_WORD	Повтор слова.
3	ERROR_CAPITALIZATION	Неверное употребление прописных и строчных букв.
4	ERROR_TOO_MANY_ERRORS	Текст содержит слишком много ошибок. При этом приложение может отправить Яндекс.Спеллеру оставшийся непроверенным текст в следующем запросе.

См. также

[Метод checkText](#)

[Метод checkTexts](#)

JavaScript API

JavaScript API предназначен для подключения Яндекс.Спеллера к клиентской HTML-странице. Интерфейс реализован в файле `spell.js` и состоит из единственного класса [Speller](#).

Для более подробных сведений о подключении API см. [Подключение к веб-формам](#)

Класс Speller

Класс `Speller` позволяет запустить проверку правописания, показать диалог настроек Яндекс.Спеллера, а также содержит средства для управления параметрами сервиса.

Константы

Имя	Значение (десятичное)	Описание
<code>IGNORE_UPPERCASE</code>	1	Пропускать слова, написанные заглавными буквами, например, "ВПК".
<code>IGNORE_DIGITS</code>	2	Пропускать слова с цифрами, например, "авп17х4534".
<code>IGNORE_URLS</code>	4	Пропускать интернет-адреса, почтовые адреса и имена файлов.
<code>FIND_REPEAT</code>	8	Подсвечивать повторы слов, идущие подряд. Например, "я полетел <i>на на</i> Кипр".
<code>IGNORE_LATIN</code>	16	Пропускать слова, написанные латиницей, например, "madrid".
<code>FLAG_LATIN</code>	128	Отмечать слова, написанные латиницей, как ошибочные.

Конструктор

Имя	Описание
<code>Speller(args)</code>	Задаёт параметры Яндекс.Спеллера. Единственным параметром конструктора является объект <code>args</code> со следующими полями:

Поля

Имя	Описание
<code>url</code>	URL каталога, в который установлен Яндекс.Спеллер, см. шаг 1 в инструкции по подключению Веб-Спеллера.
<code>lang</code>	Язык словаря проверки орфографии. Возможные значения: <ul style="list-style-type: none"> “ru” (русский); “uk” (украинский); “en” (английский). Значение по умолчанию: “ru” (русский).
<code>options</code>	Сумма значений констант , соответствующих выбранным опциям Веб-Спеллера. Значение по умолчанию: 0 (ничего не выбрано).
<code>spellDlg</code>	Размеры окна «Проверка правописания» {width,height}.
<code>optDlg</code>	Размеры окна «Параметры Яндекс.Спеллера» {width,height}.
<code>userDicDlg</code>	Размеры окна «Редактирование словаря пользователя» {width,height}.

Пример:

Создает объект `speller` с заданными настройками.

```
var speller = new Speller(
  { url: "/speller",
    lang: "ru", options: Speller.IGNORE_LATIN + Speller.IGNORE_URLS,
    spellDlg: { width: 500, height: 320 }
  }
);
```

Параметры `lang` и `options` задают «начальные» параметры Спеллера. Их значения могут быть изменены пользователем в диалоговом окне «Параметры». Введенные значения сохраняются в файле cookie для дальнейшего использования.

Размеры диалоговых окон задаются объектом с полями `width` и `height` (соответственно, ширина и высота окна в пикселах). Размеры окон – это единственные параметры пользовательского интерфейса, которые возможно изменять средствами API. Все прочие параметры изменяются посредством редактирования HTML- и CSS-файлов.

Методы

Имя	Описание
<code>check(ctrls)</code>	Показывает диалоговое окно «Проверка правописания» и начинает проверять правописание в указанных полях.
<code>optionsDialog()</code>	Показывает диалоговое окно «Параметры», в котором пользователь может выбрать язык словаря и изменить настройки Яндекс.Спеллера.

Подробное описание методов**check**

```
check(ctrls)
```

Показывает диалоговое окно «Проверка правописания» и начинает проверять правописание в указанных полях.

Параметры:

Имя	Тип	Описание
<code>ctrls</code>	<code>HTMLElement[]</code>	Массив текстовых полей, которые необходимо проверить.

Пример:

Проверка правописания в элементах `subj` и `message` HTML-формы `myForm`:

```
speller.check( [ myForm.subj, myForm.message ] );
```

optionsDialog

```
optionsDialog()
```

Показывает диалоговое окно «Параметры», в котором пользователь может выбрать язык словаря и изменить настройки Яндекс.Спеллера.

Подключение к веб-формам

Как подключить и использовать приложение Веб-Спеллер для проверки правописания на страницах сайтов.

Веб-Спеллер предназначен для проверки правописания в HTML-элементах `<textarea>` и `<input type="text">` и представляет собой полнофункциональное веб-приложение с диалоговым пользовательским интерфейсом, которое можно встроить в любую веб-страницу.

Веб-Спеллер состоит из двух частей – клиентской и серверной:

1. Клиентская часть включает три диалоговых окна (проверки правописания, настроек и редактирования пользовательского словаря) и скрипт `spell.js`. Клиентская часть должна быть установлена на том же сервере, где размещается сайт.

HTML-код клиентской части доступен для загрузки по адресу:

```
http://speller.yandex.net/speller/1.0/spell-1.0.zip
```

2. Серверная часть – это XML-сервис проверки орфографии Яндекса, см. [Web Service API](#).

Совет:

Пример подключения Веб-Спеллера можно посмотреть на странице <http://speller.yandex.net/speller/1.0/index.html>

Для того чтобы подключить Веб-Спеллер, выполните следующие шаги:

1. Загрузите архив [spell-1.0.zip](#) и распакуйте его в каталог `/speller` вашего web-приложения.

В архиве находятся все необходимые для демонстрации возможностей Веб-Спеллера файлы:

- `spell.css` — файл со стилями;
- `spell.js` — JavaScript-файл клиентской части Веб-Спеллера;
- `spelldlg.html` — диалоговое окно проверки правописания;
- `spellopt.html` — диалоговое окно настроек Веб-Спеллера;
- `userdic.html` — диалоговое окно редактирования пользовательского словаря (работает в Internet Explorer 8 и Firefox 3);
- `test.html` — простая HTML-страница с формой для ввода текста и двумя кнопками для вызова диалоговых окон.

2. На страницу с проверяемым текстом добавьте клиентский скрипт:

```
<script type="text/javascript" src="speller/spell.js"></script>
```

и функцию `spellCheck()`:

```
<script type="text/javascript">
var speller = new Speller({ url:"speller", lang:"ru",
options:Speller.IGNORE_URLS });

function spellCheck() {
    var form = document.forms["myform"];
    speller.check([ form.ctrl_1, form.ctrl_2, ..., form.ctrl_N ]);
}
</script>
```

Примечание:

Параметр `url: "speller"` задает URL каталога, в который был распакован архив (см. [пункт 1](#)).

3. Добавьте кнопки вызова диалоговых окон для проверки правописания и настроек Веб-Спеллера:

```
<button name="cmdSpell" type="button" onclick="spellCheck()">Проверить право-
писание</button>
```

```
<button type="button" onclick="speller.optionsDialog()"> Параметры...</button>
```

Для проверки работоспособности приложения наберите в поле ввода произвольный текст (разумеется, с ошибками) и нажмите на кнопку "Проверить правописание".

Совет:

Веб-Спеллер поддерживает HTTPS-протокол. Для работы по HTTPS требуется:

1. Во всех клиентских HTML-файлах заменить `http:` на `https:`
2. В файл `spelldlg.html` вставить следующий скрипт:

```
<script type="text/javascript">
Speller.URL="https://speller.yandex.net/services/spellservice.js"
</script>
```

Как изменить настройки по умолчанию?

Для того чтобы изменить настройки Веб-Спеллера по умолчанию, следует задать поля `lang` и `options` при создании объекта `Speller`.

Например, чтобы установить украинский язык проверки и игнорировать слова с прописными буквами, добавьте следующий JavaScript:

```
<script type="text/javascript">
  var speller = new Speller({ url: "speller",
    lang: "uk",
    options: Speller.IGNORE_UPPERCASE });
  ...
</script>
```

Подключение к текстовым редакторам

Как подключить Яндекс.Спеллер к популярным текстовым редакторам для веб.

Яндекс.Спеллер может быть использован как альтернатива встроенным в текстовые редакторы средствам проверки орфографии. Ниже приведены примеры использования Спеллера в следующих приложениях:

- [TinyMCE](#)
- [Drupal](#)
- [Joomla!](#)
- [WordPress](#)
- [FCKEditor 2.6](#)
- [Aspell/IsPELL](#)

Подключение к TinyMCE

Как подключить Яндекс.Спеллер к HTML-редактору TinyMCE.

TinyMCE — это кроссплатформенный WYSIWYG HTML-редактор с открытым кодом, который легко встраивается в любую HTML-страницу.

Чтобы подключить Яндекс.Спеллер к TinyMCE, выполните следующие действия:

1. Добавьте в файл примера текстового процессора (/tinymce/examples/word.html) скрипт инициализации Яндекс.Спеллера:

```
<script type="text/javascript">
  tinyMCE.init({
    // General options
    mode : "textareas",
    theme : "advanced",
    plugins : "...,spellchecker,...",
    theme_advanced_buttons3 : "...,spellchecker,...",
    // Spellchecker
    spellchecker_languages : "+Russian=ru,Ukrainian=uk,English=en",
    spellchecker_rpc_url : "http://speller.yandex.net/services/tinyspell",
    spellchecker_word_separator_chars : '\\s!"#$%&()*+,-./:;<=>@[\\]^_{}|}
\\xa7\\xa9\\xab\\xae\\xb1\\xb6\\xb7\\xb8\\xbb\\xbc\\xbd\\xbe\\u00bf\\xd7\\xf7\\xa4\\u201d
\\u201c'});
</script>
```

2. Переадресуйте запросы к спеллеру с вашего сайта на Яндекс.

На шаге 1 мы настроили TinyMCE для выполнения кросс-доменных запросов с вашего сайта на сайт Яндекса. Такое решение работает в большинстве современных браузеров (Chrome, Firefox 5, Opera 11, Safari 5). Если ваш сайт рассчитан на работу в таких браузерах – можно сразу переходить к [шагу 3](#).

Чтобы спеллер работал со всеми браузерами – потребуется переадресовать запросы спеллера сначала на ваш сайт, а потом – с вашего сайта на [Яндекс](#). Для этого в файле words.html в параметре spellchecker_rpc_url укажите URL на вашем сайте, например, так:

```
spellchecker_rpc_url : "/speller/tinyspell",
```

а затем, переадресуйте запросы от `/speller/tinyspell` на <http://speller.yandex.net/services/tinyspell>. Сделать это можно различными способами.

Например, если используется web-сервер Apache, то настроить переадресацию можно в файле `httpd.conf` с помощью инструкции `ProxyPass`:

```
ProxyPass /speller/tinyspell http://speller.yandex.net/services/tinyspell
```

Кроме того, для переадресации можно воспользоваться скриптами `tinyspell.*` из архива серверных скриптов [server-scripts.zip](#). Разместите скрипт (например, `tinyspell.php`) на вашем сайте и укажите его URL в параметре `spellchecker_rpc_url` : `"/tinyspell.php"`.

3. Подключение Яндекс.Спеллера завершено.

После обновления страницы `word.html` появится кнопка проверки правописания. Если ее нажать, слова с ошибками будут подчеркнуты красной волнистой линией. Если щелкнуть левой кнопкой мыши по слову с ошибкой, то в контекстном меню будут предложены подсказки с вариантами правильного написания.

Совет:

С помощью параметра `spellchecker_word_separator_chars` дефис исключается из списка используемых в редакторе символов-разделителей, что позволяет проверять слова с дефисом как единое целое, например: "как-нибудь", "женщина-врач", "Рио-де-Жанейро", и т.п.

Совет:

При подключении к TinyMCE [настройки Яндекс.Спеллера](#) можно передавать в URL-параметре `options`. Так, например, в запросе, отправленном на URL

```
http://speller.yandex.net/services/tinyspell?options=514
```

будут проигнорированы ошибки капитализации и не будут проверяться слова с цифрами (514 — это сумма опций `IGNORE_CAPITALIZATION` и `IGNORE_DIGITS`).

Подключение к Drupal

Как подключить Яндекс.Спеллер к системе управления сайтом Drupal.

Drupal — (Друпал) — это система управления сайтом (CMS), а также среда для создания веб-приложений (CMF). По умолчанию, в Drupal отсутствует визуальный редактор, однако существует удобный модуль **Wysiwyg**, который позволяет без труда подключать к Drupal популярные WYSIWYG-редакторы.

О том, как установить модуль Wysiwyg, можно прочитать в документации на страницах проекта или посмотреть [видео](#).

Drupal и TinyMCE

Допустим, вы решили подключить к Drupal популярный редактор TinyMCE. В TinyMCE встроен конфигурируемый плагин проверки орфографии (процедура его конфигурации для работы с Яндекс.Спеллером в общем случае описана на странице [Подключение к TinyMCE](#)), однако доступ к нему в визуальном конфигураторе модуля Wysiwyg отсутствует.

Чтобы это исправить, выполните следующие действия:

1. Откройте файл конфигурации редактора `tinymce.inc`, расположенный в директории `/sites/all/modules/wysiwyg/editors`, и добавьте следующий фрагмент кода в самый конец файла (до строки `return $plugins`):

```
// start YANDEX block
if (version_compare($editor['installed version'], '3.4.6', '>=')) {
  $plugins['spellchecker'] = array(
    'path' => $editor['library path'] . '/plugins/spellchecker',
    'buttons' => array('spellchecker' => t('Spellchecker')),
    'options' => array(
      'spellchecker_languages' => "+Russian=ru,Ukrainian=uk,English=en",
      'spellchecker_rpc_url' => "/drupal/sites/all/libraries/tinymce/
tinyspell.php",
      'spellchecker_word_separator_chars' => ("\\s!\"#$%&()*+.,/:;<=>?
@[\\]^_{}|}\xc2\xa7 \xc2\xa9" .
      "\\xc2\xab\xc2\xae
\\xc2\xba1\\xc2\xbb6\\xc2\xbb7\\xc2\xbb8\\xc2\xbbb\\xc2\xbc\\xc2\xbd" .
      "\\xc2\xbe\\xc2\xbf
\\xc3\x97\\xc3\x97\\xc2\xa4\xe2\x80\x9d\xe2\x80\x9c" )
    ),
    'url' => 'http://wiki.moxiecode.com/index.php/TinyMCE:Plugins/
spellchecker',
    'internal' => TRUE,
    'load' => TRUE,
  );
}
// finish YANDEX block
return $plugins;
```

Обратите внимание, что в данном случае значение параметра `spellchecker_rpc_url` равно `/drupal/sites/all/libraries/tinymce/tinyspell.php`, чтобы все, связанное с TinyMCE, хранить в одном месте. Соответственно, настраивать переадресацию или размещать вспомогательный скрипт нужно будет именно по этому адресу. Подробнее про настройку переадресации можно прочитать в разделе [Подключение к TinyMCE](#).

2. Откройте визуальный configurator TinyMCE в административной панели (**Modules > Wysiwyg | Configure > TinyMCE | Edit > Buttons and Plugins**) и установите флажок **Spell Checker**. Сохраните изменения — соответствующая кнопка появится в интерфейсе TinyMCE на страницах, где можно редактировать текст.

В результате в редакторе должна заработать проверка правописания для русского языка.

Drupal и CKEditor

Установка и настройка WYSIWYG-редактора [CKEditor](#) во многом схожа с установкой TinyMCE. Проверьте, что после установки редактор находится в директории `/sites/all/libraries/ckeditor`.

Для подключения Яндекс.Спеллера выполните следующие действия:

1. [Скачайте](#) и распакуйте содержимое архива `aspell` в директорию `ckeditor/plugins` (в результате должна появиться новая директория `aspell`, подробнее см. [Подключение к CKEditor](#)).
2. Откройте файл конфигурации `ckeditor.inc` в директории `/sites/all/modules/wysiwyg/editors` и внесите следующие изменения:
 - 2.1. Добавьте плагин `aspell` к доступным плагинам редактора. Для этого найдите строку инициализации массива `$extra_plugins` внутри функции `wysiwyg_ckeditor_settings` и добавьте строку `'aspell'` в этот массив:

```
$settings['toolbar'] = array();
if (!empty($config['buttons'])) {
    $extra_plugins = array('aspell'); // YANDEX: 'aspell' добавлен в список
    доступных плагинов
    // ...
}
```

- 2.2. Добавьте кнопку проверки с помощью плагина aspell (кнопка **SpellCheck**, название определено в описании плагина) в визуальный configurator. Для этого внесите изменения в массив `buttons` в функции `wysiwyg_ckeditor_plugins` (чтобы не путаться, можно закомментировать стандартные кнопки проверки орфографии):

```
'Maximize' => t('Maximize'),
// YANDEX: закомментированы стандартные кнопки проверки орфографии
// 'SpellChecker' => t('Check spelling'), 'Scayt' => t('Check spelling as
you type'),
'SpellCheck' => t('Spell Checker'), // YANDEX: кнопка плагина aspell
добавлена в configurator
'About' => t('About'),
```

- Откройте визуальный configurator FCEditor в административной панели (**Modules -> Wysiwyg | Configure -> CKEditor | Edit -> Buttons and Plugins**) и установите флажок **Spell Checker**.
- Сохраните изменения.

Кнопка проверки орфографии появится в интерфейсе CKEditor на страницах, где можно редактировать текст.

См. также

[Яндекс.Спеллер как замена Aspell/Ispell](#)

Подключение к CKEditor

Как подключить Яндекс.Спеллер к редактору CKEditor.

CKEditor — это текстовый WYSIWYG редактор с открытым кодом для использования на веб-страницах.

Чтобы подключить Яндекс.Спеллер к CKEditor необходимо установить плагин `aspell`, подключающий библиотеку [Speller Pages](#), и сконфигурировать его для работы с сервисом Яндекс.Спеллер. Архив с уже сконфигурированным плагином можно скачать с [нашего сайта](#). Ссылка на оригинал плагина `aspell` доступна на соответствующей [странице](#) обсуждения в форуме разработчиков.

Для подключения Яндекс.Спеллера:

- Распакуйте содержимое архива `aspell` в директорию `ckeditor/plugins` (в результате должна появиться новая директория `aspell`).
- Подключите `aspell` к CKEditor в файле конфигурации (`ckeditor/config.js`) и добавьте кнопку проверки орфографии **SpellCheck** на панель инструментов редактора. В следующем примере показано, как подключается плагин и добавляется кнопка **SpellCheck** вместо стандартных кнопок проверки орфографии. Кнопка добавляется на панель **Full**, которая используется по умолчанию:

```
CKEDITOR.editorConfig = function( config )
{
    config.extraPlugins += (config.extraPlugins ? ',aspell' : 'aspell' );
    config.toolbar_Full[2] = { name: 'editing', items :
[ 'Find', 'Replace', '-', 'SelectAll', '-', 'SpellCheck' ] };
};
```

Примечание:

Подробнее о настройке панелей инструментов можно прочитать на [страницах документации](#) CKEditor.

Совет:

Директория `ckeditor/_samples/` содержит подробные примеры подключения сконфигурированного редактора на страницы сайта.

По умолчанию плагин использует скрипт `aspell/spellerpages/server-scripts/spellchecker.php` для получения данных с сервиса Яндекс.Спеллер, требующий настроенного интерпретатора PHP. Для того, чтобы переключиться на версию скрипта на языке Perl, нужно изменить путь к скрипту в файле `aspell/dialogs/aspell.js`:

```
oSpeller.spellCheckScript = editor.plugins.aspell.path+'spellerpages/server-scripts/spellchecker.pl; // YANDEX: расширение 'php' изменено на 'pl'
```

См. также

[Яндекс.Спеллер как замена Aspell/IsPELL](#)

Подключение к Joomla!

Как подключить Яндекс.Спеллер к системе управления сайтом Joomla!.

Joomla! — это система управления сайтом с открытым кодом. В качестве встроенного редактора в Joomla! используется TinyMCE, к которому можно подключить Яндекс.Спеллер. Но проще это сделать через компонент **JCE**, который уже содержит средства проверки правописания.

Чтобы подключить Яндекс.Спеллер к JCE, скачайте архив с патчем [jce.1.5.zip](#) или [jce.2.0.zip](#) (в соответствии с вашей версией Joomla!) и выполните действия, описанные в файле архива README.txt.

Подключение к WordPress

Как подключить Яндекс.Спеллер к системе управления контентом WordPress.

WordPress — это система управления контентом с открытым кодом, распространяемая под лицензией GNU GPL.

Чтобы подключить Яндекс.Спеллер к WordPress, используйте один из следующих плагинов, разработанных пользователями API:

- [Q2W3 Yandex Speller](#);
- [Yandex Speller Application](#)

В результате в редакторе должна заработать проверка правописания для русского языка.

Подключение к FCKEditor 2.6

Как подключить Яндекс.Спеллер к редактору FCKEditor.

FCKEditor — это текстовый WYSIWYG редактор с открытым кодом для использования на веб-страницах..

FCKEditor 2.6 поддерживает несколько интерфейсов для подключения модулей проверки правописания. Один из них — **Speller Pages** — использует программу **Aspell**. В этом разделе описывается, как подключить Яндекс.Спеллер к настроенному для работы с Aspell интерфейсу Speller Pages.

Внимание!

Серверный модуль проверки орфографии Speller Pages не является выбором по умолчанию в редакторе FCKEditor и требует от разработчиков определенной работы по настройке. Рекомендуется внимательно ознакомиться с документацией по подключению модуля, размещенной на [сайте FCK](#).

Если в FCKEditor уже настроен интерфейс Speller Pages, то подключение Яндекс Спеллера заключается в замене вызова `aspell.exe` на вызов `perl yspell.pl` (см. [шаг 3](#) ниже).

Для подключения Яндекс.Спеллера внесите следующие изменения в файлы конфигурации редактора:

1. В конфигурационном файле `fckconfig.js` укажите следующие значения для опций `FCKConfig.SpellChecker` и `FCKConfig.SpellerPagesServerScript`:

```
FCKConfig.SpellChecker = 'SpellerPages'; // 'WSC' | 'SCAYT' | 'SpellerPages'
| 'ieSpell'
FCKConfig.SpellerPagesServerScript = 'server-scripts/spellchecker.pl'; //
Available extension: .php .cfm .pl
```

2. В файле `editor/dialog/fck_spellerpages/spellerpages/spellchecker.js` замените строку, содержащую `this.spellCheckScript` на следующую:

```
this.spellCheckScript = 'speller/server-scripts/spellchecker.pl';
```

3. В скрипте `editor/dialog/fck_spellerpages/spellerpages/server-scripts/spellchecker.pl` замените вызов исполняемого модуля "`aspell.exe`" на вызов "`perl yspell.pl`" и задайте языковую отметку, например, "`ru_RU`" для проверки текстов только на русском языке или "`ru_RU,en_US`" для проверки текстов на русском и английском языках.

То есть, строки

```
my $aspell_cmd = '"C:\Program Files\Aspell\bin\aspell.exe"'; # by FredCK
(for Windows)
my $lang = 'en_US';
```

замените на

```
my $aspell_cmd = '[full-path-to]/perl.exe [full-path-to]/ypspell.pl'; # by
FredCK (for Windows)
my $lang = 'ru_RU';
```

Совет:

Скрипт `ypspell.pl` можно скачать по адресу <http://speller.yandex.net/speller/ypspell-1.0.zip>.

Примечание:

FCKeditor включает в себя реализации скрипта `server-scripts/spellchecker.pl` на **PHP** и **ColdFusion**. Для них подключение Яндекс.Спеллера осуществляется аналогично.

- После внесения изменений в файлы обновите кэш браузера.

Теперь для проверки правописания будет использоваться Яндекс.Спеллер.

См. также

[Яндекс.Спеллер как замена Aspell/Ispell](#)

Яндекс.Спеллер как замена Aspell/Ispell

Как использовать Яндекс.Спеллер вместо Aspell/Ispell.

Aspell — это свободная программа для проверки орфографии в Unix и Windows-приложениях, распространяющихся под лицензией [GNU](#).

Вместо Aspell может быть использован Яндекс.Спеллер. Для этого:

- Загрузите скрипт Яндекс.Спеллера `yspell.pl` с сайта <http://speller.yandex.net/speller/yspell-1.0.zip>.
- Для запуска проверки орфографии вместо команды "aspell" используйте команду "perl yspell.pl".

Скрипт `yspell.pl` поддерживает основные опции Aspell, поэтому вносить дополнительные изменения скорее всего не придется.

В качестве примера в руководстве рассмотрено подключение Яндекс.Спеллера к редактору [FCKeditor 2.6](#), использующему Aspell.

Как проверить, что все работает правильно

- Создайте файл `test.txt`, содержащий текст с орфографической ошибкой, например, "Текст с ашипкой".
- Перейдите в директорию, где находится скрипт `yspell.pl`.
- В командной строке выдайте следующую команду: `perl yspell.pl -encoding ENCODING < test.txt`, где вместо ENCODING укажите кодировку файла `test.txt`, например, `utf-8`, `cp1251` или `koï8`.

Если скрипт настроен верно, на экран будет выведено сообщение:

```
@ (#) Yandex Speller 1.0
*
*
& ашипкой 1 12: ошибкой
```

Интересные решения

В этом разделе собраны ссылки на интересные решения сторонних разработчиков, в которых используется API Яндекс.Спеллера.

Если вы разработали свой проект с применением API Яндекс.Спеллера и хотите предложить его пользователям, расскажите нам о нем по адресу: speller-support@yandex-team.ru.

Подключение к WordPress

[WordPress](#) — это система управления контентом с открытым кодом, распространяемая под лицензией GNU GPL.

Чтобы подключить Яндекс.Спеллер к WordPress, используйте один из следующих плагинов, разработанных пользователями API:

- [Q2W3 Yandex Speller](#);
- [Yandex Speller Application](#)

В результате в редакторе должна заработать проверка правописания для русского языка.

См. также
[Подключение к текстовым редакторам](#)

Предметный указатель

#

\$extra_plugins 18

A

args 12

ArrayOfSpellResult 9

Aspell 22

B

BY_WORDS 11

C

callback 6, 9

checkText 5

checkTexts 7

CKEditor 19

code 7, 9

col 7, 9

ColdFusion 22

ctrls 13

D

Drupal 17

E

error 7, 9

ERROR_CAPITALIZATION 11

ERROR_REPEAT_WORD 11

ERROR_TOO_MANY_ERRORS 11

ERROR_UNKNOWN_WORD 11

F

FCKConfig.SpellChecker 21

FCKConfig.SpellerPagesServerScript 21

FCKEditor 20

FIND_REPEAT 12

FIND_REPEAT_WORDS 10

FLAG_LATIN 10, 12

format 6, 9

H

height 13

html 6, 9

I

ie 4

IGNORE_CAPITALIZATION 11

IGNORE_DIGITS 10, 12

IGNORE_LATIN 10, 12

IGNORE_UPPERCASE 6, 8, 10, 12

IGNORE_URLS 6, 8, 10, 12

Ispell 22

J

Joomla 20

JSON 5–9

JSONP 5–8, 10

L

lang 6, 8, 12, 13, 15

len 7, 9

N

NO_SUGGEST 10

O

optDlg 12

options 6, 8, 12, 13, 15, 17

P

PHP 22

plain 6, 9

pos 7, 9

ProxyPass 17

R

row 7, 9

S

s 7, 9

SOAP 5

spellCheck() 14

spellchecker_rpc_url 16, 18

spellchecker_word_separator_chars 17

spellDlg 12

Speller 12

SpellResult 6, 7, 9

T

text 6, 8

TinyMCE 16

U

url 12

userDicDlg 12

W

width 13

word 7, 9

WordPress 20, 23

wysiwyg_ckeditor_settings [18](#)

Х

XML [6](#), [8](#), [9](#)

В

Веб-Спеллер [14](#)

И

интересные решения [23](#)

К

коды ошибок [11](#)

Н

настройки [10](#)

О

опции [10](#)

Я

Яндекс.Спеллер [4](#)



API Яндекс.Спеллера
Руководство разработчика

21.02.2012