

Модуль подсистемы “Пользовательские интерфейсы” <WebUser>

Модуль:	WebUser
Имя:	Web-интерфейс от пользователя
Тип:	Пользовательские интерфейсы
Источник:	ui_WebUser.so
Версия:	0.6.2
Автор:	Роман Савоченко
Описание:	Позволяет создавать собственные пользовательские web-интерфейсы на любом языке OpenSCADA.
Лицензия:	GPL

Оглавление

Модуль подсистемы “Пользовательские интерфейсы” <WebUser>	1
Введение	2
1. WEB — страницы	4

Введение

Модуль WebUser предоставляет пользователю механизм создания Web-страниц, а также позволяет обрабатывать иные Web-запросы на одном из внутренних языков OpenSCADA, обычно JavaLikeCalc, не прибегая к низкоуровневому программированию OpenSCADA.

Кроме принадлежности модуля системе OpenSCADA он также принадлежит и является модулем модуля транспортного протокола <HTTP>. Собственно, вызов WebUser производится из Protocol.HTTP. Вызов производится посредством расширенного механизма коммуникации через экспортированные в модуле WebUser функции: HttpGet() и HttpSet().

Адресация страниц начинается со второго элемента URI. Это связано с тем, что первый элемент URI используется для идентификации самого модуля пользовательского Web-интерфейса. Например URL: *http://localhost.localdomain:10002/WebUser/UserPage* можно расшифровать как вызов пользовательской страницы “UserPage” Web модуля WebUser на хосте localhost.localdomain через порт 10002. В случае отсутствия второго элемента URI и указания отображать индекс пользовательских страниц в конфигурации, формируется индекс страниц (рис.1).

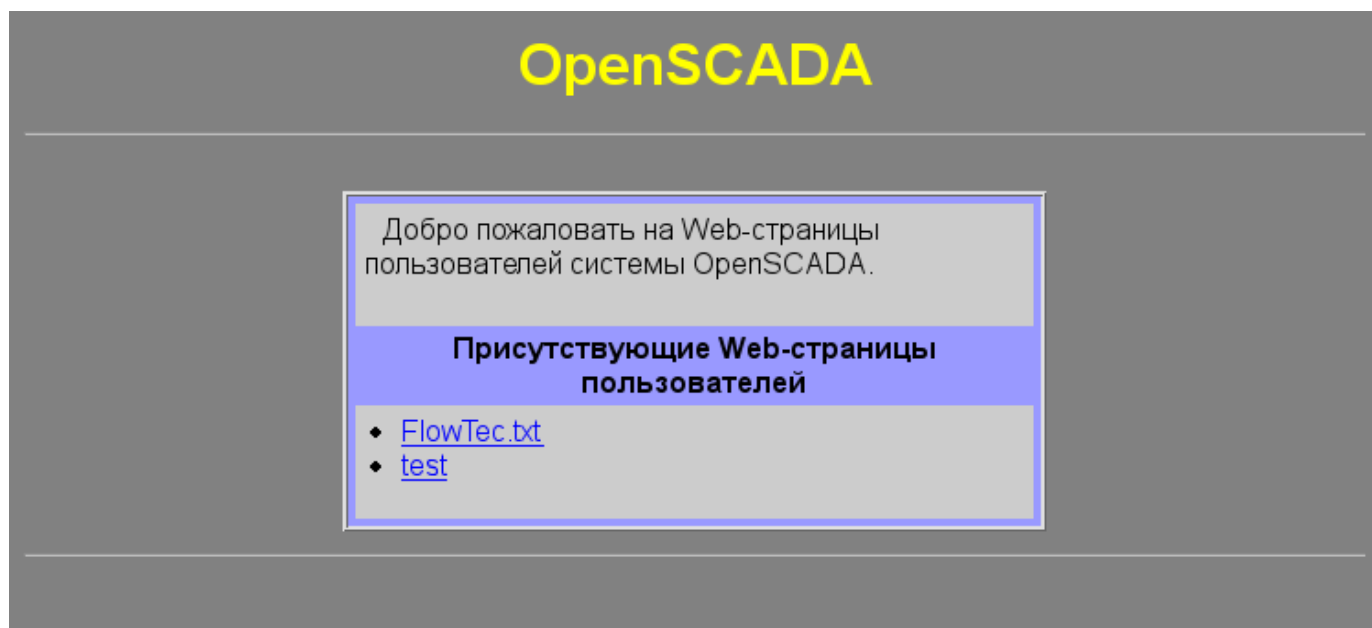


Рис.1. Индекс пользовательских страниц.

Главная вкладка конфигурации модуля (рис.2) содержит состояние модуля, предоставляет возможность выбора страницы по умолчанию и позволяет сформировать перечень пользовательских страниц.

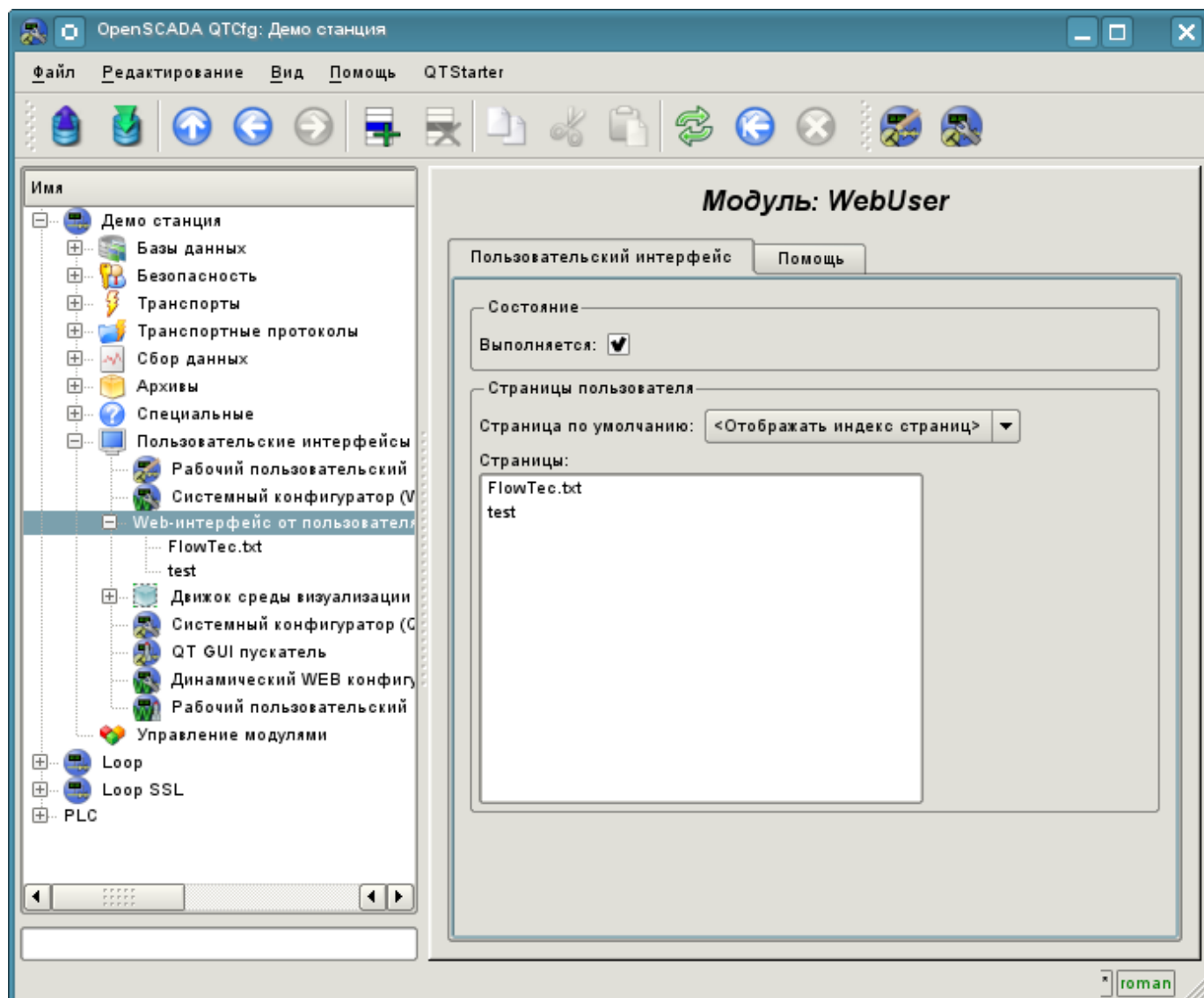


Рис.2. Основная вкладка конфигурации модуля.

1. WEB — страницы

Модуль предоставляет возможность создания реализаций множества Web-страниц в объекте "Пользовательская страница" (рис.3).

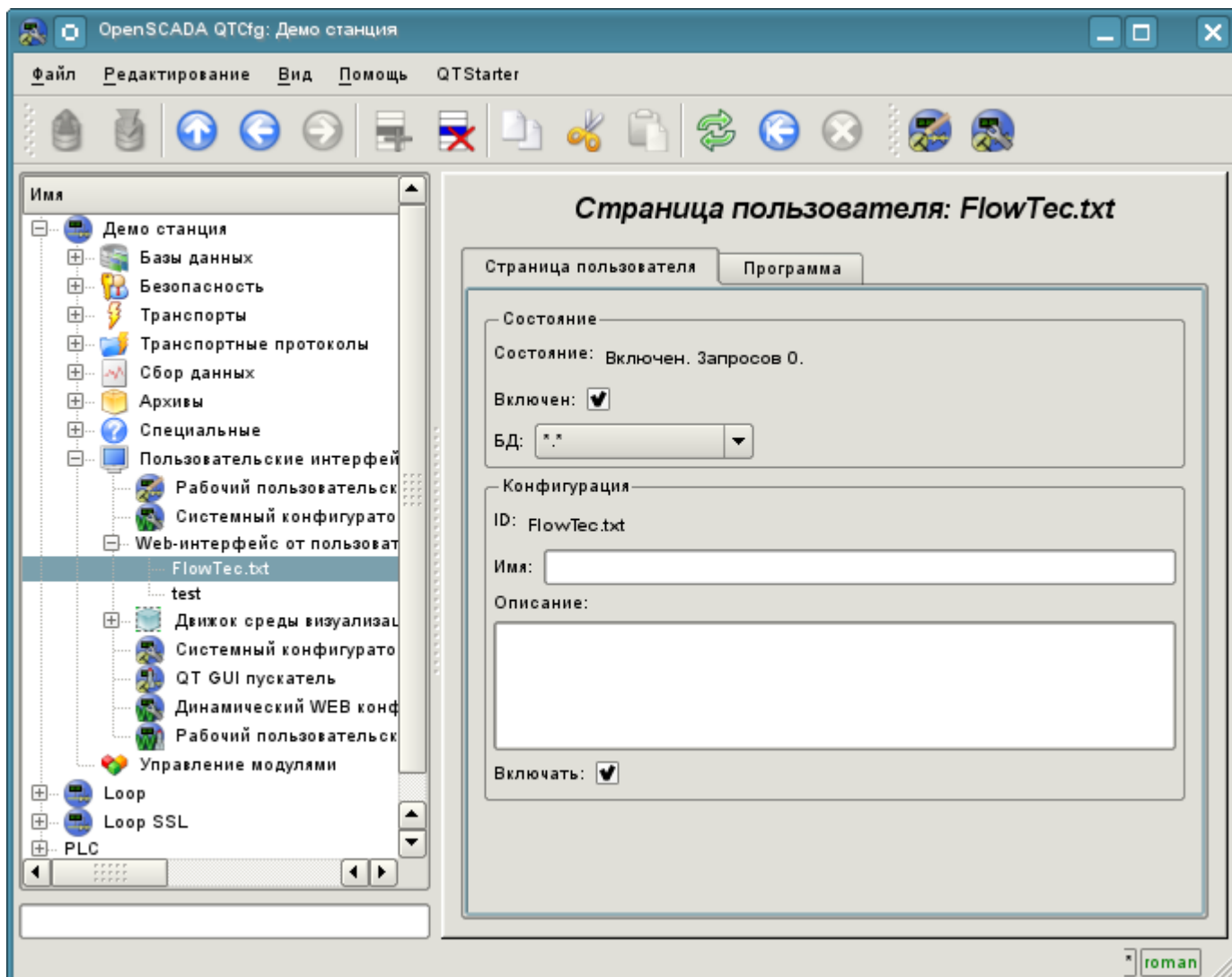


Рис.3. Главная страница конфигурации пользовательской страницы.

Главная вкладка содержит основные настройки пользовательского протокола:

- Раздел "Состояние" — содержит свойства, характеризующие состояние пользовательской страницы:
 - *Включен* — состояние страницы "Включена".
 - *БД* — БД, в которой хранится конфигурация.
- Раздел "Конфигурация" — непосредственно содержит поля конфигурации:
 - *ID* — информация об идентификаторе страницы.
 - *Имя* — указывает имя страницы.
 - *Описание* — краткое описание страницы и её назначения.
 - *Включать* — указывает на состояние "Включен", в которое переводит страницу при загрузке.

OpenSCADA QTcfg: Демо станция

Файл Редактирование Вид Помощь QTStarter

Имя

- Демо станция
 - Базы данных
 - Безопасность
 - Транспорты
 - Транспортные протоколы
 - Сбор данных
 - Архивы
 - Специальные
 - Пользовательские интерфейсы
 - Рабочий пользовательский
 - Системный конфигуризатор (V)
 - Web-интерфейс от пользователя
 - FlowTec.txt
 - test
 - Движок среды визуализации
 - Системный конфигуризатор (C)
 - QT GUI пускатель
 - Динамический WEB конфи
 - Рабочий пользовательский
 - Управление модулями
 - Loop
 - Loop SSL
 - PLC

Страница пользователя: FlowTec.txt

Страница пользователя Программа

Язык программы: JavaLikeCalc.JavaScript

Программа:

```
//SYS.messDebug("TESTWWW","HTTPreq: "+HTTPreq+. URL: "+url+. Sender: "+sender+
//for( var i in HTTPvars ) SYS.messDebug("test","Prop "+i+" = "+HTTPvars[i]);
if( cnts.length )
{
  //SYS.messDebug("TESTWWW","Content-Disposition: "+cnts[0].attr("Content-Disposition"));
  //SYS.messDebug("TESTWWW","Content-Type: "+cnts[0].attr("Content-Type"));
  dataBlk = cnts[0].text();
  if( dataBlk.length )
  {
    SYS.messDebug("TESTWWW","Text: "+dataBlk);
    SYS.messDebug("TESTWWW","Object: '"+dataBlk.parse(0,"\\n")+"'");
    for( i = 2; (sc=dataBlk.parse(i,"\\n")).length; i++ )
    {
      if( sc.slice(0,1) == "v" ) SYS.messDebug("TESTWWW","Calc: '"+sc+"'");
      else if( sc.slice(0,2) == "n" )
      {
        nd = dataBlk.parse(++i,"\\n");
        SYS.messDebug("TESTWWW",
          "Node: '"+sc.slice(1)+"': '"+nd.parse(0," ",2,true)+"'+nd.parse(1," ",2,true)+"' = '"+nd.p
          SYS.DAQ.LogicLev.experiment.F3.ed.set(nd.parse(3," ",2,true));
      }
    }
  }
}
//SYS.messDebug("TESTWWW","Page: "+page);
```

roman

Вкладка процедуры обработки запросов к пользовательской странице содержит поле для выбора внутреннего языка программирования OpenSCADA и поле ввода текста процедуры обработки.

- *rez* — Результат обработки (по умолчанию — "200 OK").
- *HTTPreq* — Метод HTTP запроса (GET,POST).
- *url* — URI запроса.
- *page* — Содержимое страницы Get/Post, как для запроса так и для ответа.
- *sender* — Отправитель запроса.
- *user* — Аутентифицированный пользователь.
- *HTTPvars* — HTTP переменные в Object. Изменённые и добавленные переменные (кроме "Date", "Server", "Accept-Ranges" и "Content-Length") будут помещены в пакет ответа.
- *URLprms* — Параметры URL в Object.
- *cnts* — Элементы содержимого для POST в Array<XMLNodeObj>.

Общий сценарий запроса пользовательской страницы:

- Внешняя сетевая станция формирует HTTP запрос с URI вида `"/WebUser/<UserPage>"`, который попадает на транспорт OpenSCADA с значением поля конфигурации "Протокол", равного "HTTP".
- Транспорт направляет запрос на модуль транспортного протокола `Protocol.HTTP`.
- Модуль транспортного протокола в соответствии с первым элементом URI направляет запрос данному модулю.
- Данный модуль выбирает объект страницы пользователя, которая указана во втором элементе URI.
- Выполняется инициализация переменных HTTP-протокола для процедуры страницы:
 - *HTTPreq* — устанавливается в значение строки "GET" или "POST" в зависимости от типа запроса;
 - *url* — адрес запрашиваемого ресурса (URI);
 - *page* — содержимое передаваемой страницы для метода "POST";
 - *sender* — адрес отправителя запроса;
 - *user* — адрес аутентифицированного пользователя, если аутентификация имела место;
 - *HTTPvars* — разобранный перечень переменных протокола HTTP в виде свойств объекта;
 - *URLprms* — разобранный перечень параметров URL в виде свойств объекта;
 - *cnts* — разобранные элементы содержимого для POST в `Array<XMLNodeObj>` с содержимым элементов в тексте и свойствами в атрибутах `XMLNodeObj`.
- Вызов процедуры на исполнение, которая, обработав запрос, формирует содержимое страницы в "page" и результат запроса в "rez".
- В завершение формируется ответ с кодом возврата HTTP из "rez" и содержимым из "page", а также изменёнными и добавленными переменными протокола HTTP из *HTTPvars*.