qooxdoo

Software Engineering II
Introduction

- JavaScript framework
- Focus: Stand-alone browser-based applications
- Comprehensive GUI toolkit
- Advanced client-server communication
- Open Source (LGPL / EPL)
- Initiated by 1&1 Internet AG in 2005

http://qooxdoo.org
Requirements (End Users)

😊 Common web browser
- Internet Explorer 6+
- Firefox 2+
- Opera 9+
- Safari 3+
- Chrome 2+

😊 No browser extensions required
(no ActiveX, Java, Flash, Silverlight, etc.)
Requirements (Developers)

😊 No HTML
😊 No DOM
😊 No CSS
😊 No specific server-side software
😊 Develop on any platform
😊 Object-oriented programming
😊 JavaScript
GUI Toolkit

- Many standard widgets
- Simple creation of custom widgets
- Powerful layout managers
- Full keyboard support
- Drag & drop
- Event-based programming
- Real-world theming
Demo

- Showcase
Theming

- Decoupled from widget code
- Simple yet powerful configuration files
- No CSS means no CSS hacks
// create a button
var button = new qx.ui.form.Button("First Button",
   ".resource/image/test.png");

// provide size hints and/or min/max
button.setWidth(120);
button.setMinWidth(80);

// add it to the document
this.getRoot().add(button, { top: 10, left: 10 });

// action upon click/Enter
button.addListener("execute",
   function(e)
   {
       alert("Hello World!");
   });
qx.Class.define("custom.Application",
{
    extend : qx.application.Gui,       // extend the Gui class

    members :
    {
        main : function()          // override main()
        {
            this.base(arguments);     // call superclass' main()

            var button = new qx.ui.form.Button("First Button",
                "/resource/image/test.png");
            button.setWidth(120);
            button.setMinWidth(80);

            this.getRoot().add(button, { top: 10, left: 10 });

            button.addListener("execute",
                function(e) { alert("Hello World!"); } );
        }
    }
})
Event-based Programming

- Property value changes
- Abstraction layer directly above DOM
  - “execute”, “beforeDisappear”, etc.
- Unified key event handling
  - “keydown”, “keypress”, “keyinput”, “keyup”
- Unified mouse event handling
  - mouse clicks, mouse wheel, drag & drop
- Custom events
- Generic DOM events
Build Process

Application development (./generate.py source)
- Checks, Debugging

Application deployment (./generate.py build)
- Compressor (for minimum size)
- Optimization (for maximum performance)
- Dependencies (with automatic resolving)
- Modules, Packages (for optimal distribution)
- Target directories (for optimal integration)
Object-oriented JavaScript

- Fully class-based
- Namespaces
- Only noncritical manipulation of native objects
- Static classes, abstract classes, singletons
- Interfaces, mixins
- Public, protected, private members
- Dynamic properties
qx.Class.define("qx.ui.form.Button", {
    extend : qx.ui.basic.Atom,
    implement : [qx.IClickable, ...],
    include : [qx.MToggle, ...],
    construct : function() { ... },
    properties : { ... },
    statics : { ... },
    members : { ... },
    environment: { ... },
    events : { ... },
    destruct : { ... },
    defer : { ... }
});
Communication with Server

- **Generic transport layer**
  - Queues, Timeouts
  - synchronous, async, cross-domain, file upload, form fields
  - Data exchange
    - via JSON, XML, JavaScript, ...

- **Not just wrapper around XMLHttpRequest**
  - Multiple transports available
Remote Procedure Calls (RPC)

```javascript
var rpc = new qx.io.remote.Rpc();
rpc.setUrl("/services/");
rpc.setServiceName("qooxdoo.test");

var callback = function(result, ex, id)
{
    if (ex == null)
        alert("result: " + result);
    else
        alert("exception: " + ex);
}
rpc.callAsync(callback, "echo", "Hello world!");

function method_echo($params, $error)
{
    if (count($params) != 1)
    {
        $error->SetError(JsonRpcError_ParameterMismatch,
                        "Expected 1 parameter; got " . count($params));
        return $error;
    }
    return "Client said: [" . $params[0] . "]";
}
```
API Reference

- Automatic generation (./generate.py api)
- Full-featured JavaScript parser
- Javadoc-like comments
- Also for documenting a custom application!
- Searchable

http://api.qooxdoo.org
Demo

- API Viewer
Have you backed up your work?

Our Disaster Recovery Plan Goes Something Like This...

HELP! HELP!