Declarative UI Language Tutorial

Introduction

This tutorial walks through common template patterns in the declarative UI language, showing how templates compile to HTMLNode trees and integrate with server-side backing objects through two-way data bindings.

1. Basic Template Structure

Simple Text and Containers

javascript

```
// Basic page structure
page("Welcome") {
  header { "My Application" }
  main {
    section { "Welcome to our platform!" }
    paragraph { "This is a simple text paragraph." }
  }
  footer { "© 2025 My Company" }
}
```

Compiles to HTMLNode Tree:

javascript

```
HTMLNode.element("div", { class: "page" }, [
  HTMLNode.element("header", {}, [HTMLNode.text("My Application")]),
  HTMLNode.element("main", {}, [
    HTMLNode.element("section", {}, [HTMLNode.text("Welcome to our platform!")]),
    HTMLNode.element("p", {}, [HTMLNode.text("This is a simple text paragraph.")])
  ]),
  HTMLNode.element("footer", {}, [HTMLNode.text("© 2025 My Company")])
])
```

Layout Containers

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Common layout patterns
dashboard {
 row {
  column(width: 3) {
   sidebar {
    nav { "Navigation Menu" }
    menu {
     item { "Dashboard" }
     item { "Users" }
     item { "Reports" }
    }
   }
  }
  column(width: 9) {
   content {
    header { "Main Content Area" }
    section { "Content goes here..." }
   }
  }
 }
}

// Grid layouts
grid(columns: 3, gap: "1rem") {
 card { "Item 1" }
 card { "Item 2" }
 card { "Item 3" }
 card { "Item 4" }
 card { "Item 5" }
 card { "Item 6" }
}

// Flexbox layouts
flex(direction: "row", justify: "space-between") {
 div { "Left content" }
 div { "Center content" }
 div { "Right content" }
}
```

2. Lists and Iteration

Simple Lists

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Static list
list {
  item { "First item" }
  item { "Second item" }
  item { "Third item" }
}

// Unordered list with styling
ul(style: "bullet") {
  li { "Apple" }
  li { "Banana" }
  li { "Cherry" }
}

// Ordered list
ol {
  li { "Step one: Login" }
  li { "Step two: Navigate to dashboard" }
  li { "Step three: Start working" }
}
```

Dynamic Lists with Server Data

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// List bound to server-side collection
// users is a server-side backing object with two-way binding
list {
 for (user in users) {
  listItem {
   avatar { src: user.profileImage }
   content {
    title { user.name }
    subtitle { user.email }
    badge { user.status }
   }
   actions {
    button("Edit") { onclick: editUser(user) }
    button("Delete") { onclick: deleteUser(user) }
   }
  }
 }
}

// Card-based list layout
cardList {
 for (product in products) {
  card {
   image { src: product.imageUrl, alt: product.name }
   header { product.name }
   body {
    text { product.description }
    price { product.price }
   }
   footer {
    button("Add to Cart") { onclick: addToCart(product) }
    button("View Details") { onclick: viewProduct(product.id) }
   }
  }
 }
}
```

3. Forms and Input Controls

Basic Form Elements

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Simple contact form
form {
 fieldset("Contact Information") {
  field {
   label { "Full Name" }
   input(type: "text", bind: contact.name, required: true)
  }

  field {
   label { "Email Address" }
   input(type: "email", bind: contact.email, required: true)
  }

  field {
   label { "Phone Number" }
   input(type: "tel", bind: contact.phone)
  }

  field {
   label { "Message" }
   textarea(bind: contact.message, rows: 4)
  }
 }

 actions {
  button("Submit") { type: "submit", onclick: submitContact(contact) }
  button("Cancel") { type: "button", onclick: cancelForm() }
 }
}
```

Checkboxes and Radio Buttons

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Checkbox groups
form {
 fieldset("Preferences") {
  field {
   label { "Notification Settings" }
   checkboxGroup(bind: user.notifications) {
    checkbox("email") { "Email notifications" }
    checkbox("sms") { "SMS notifications" }
    checkbox("push") { "Push notifications" }
   }
  }

  field {
   label { "Account Type" }
   radioGroup(bind: user.accountType) {
    radio("basic") { "Basic Account" }
    radio("premium") { "Premium Account" }
    radio("enterprise") { "Enterprise Account" }
   }
  }
 }
}

// Individual checkboxes with binding
checkbox {
 bind: user.agreeToTerms
 label: "I agree to the terms and conditions"
 required: true
}

toggle {
 bind: settings.darkMode
 label: "Enable dark mode"
}
```

Select Dropdowns and Options

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Simple dropdown
field {
  label { "Country" }
  select(bind: user.country) {
    option(value: "us") { "United States" }
    option(value: "ca") { "Canada" }
    option(value: "uk") { "United Kingdom" }
    option(value: "de") { "Germany" }
  }
}

// Dynamic dropdown from server data
field {
  label { "Department" }
  select(bind: user.departmentId) {
    for (dept in departments) {
      option(value: dept.id) { dept.name }
    }
  }
}

// Multi-select
field {
  label { "Skills" }
  multiSelect(bind: user.skills) {
    for (skill in availableSkills) {
      option(value: skill.id) { skill.name }
    }
  }
}
```

4. Two-Way Data Binding with Server Objects

Object Binding Fundamentals

Copyright @2024 ZBlox. All Rights Reserved

```javascript
// Server-side backing object (conceptual)
// This object exists on the server and is synchronized with client
class User {
  id: number
  name: string
  email: string
  preferences: UserPreferences
}

// Client template with two-way binding
form(bind: user) {
  field {
    label { "Name" }
    input(bind: user.name) // Changes sync back to server object
  }

  field {
    label { "Email" }
    input(bind: user.email) // Changes sync back to server object
  }

  field {
    label { "Theme" }
    select(bind: user.preferences.theme) {
      option("light") { "Light Theme" }
      option("dark") { "Dark Theme" }
    }
  }
}
```

Binding Synchronization

When user types in the name field:

Client Update:

Copyright @2024 ZBlox. All Rights Reserved

json

```
{
  "type": "property_update",
  "objectId": "user-123",
  "property": "name",
  "newValue": "John Smith",
  "oldValue": "John Doe",
  "timestamp": "2024-12-15T10:30:00Z"
}
```

Server Response:

json

```
{
  "type": "property_acknowledged",
  "objectId": "user-123",
  "property": "name",
  "value": "John Smith",
  "validationErrors": null
}
```

Validation and Error Handling

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Form with validation binding
form(bind: user) {
  field {
    label { "Email" }
    input(bind: user.email)
    if (user.validationErrors.email) {
      errorMessage { user.validationErrors.email }
    }
  }

  field {
    label { "Age" }
    input(type: "number", bind: user.age, min: 18, max: 120)
    if (user.validationErrors.age) {
      errorMessage { user.validationErrors.age }
    }
  }
}
```

Collection Binding

javascript

```
// List bound to server collection
list(bind: users) {
  for (user in users) {
    listItem {
      input(bind: user.name) // Individual item binding
      checkbox(bind: user.active)
      button("Remove") { onclick: users.remove(user) } // Collection method
    }
  }

  actions {
    button("Add User") { onclick: users.add(new User()) } // Collection method
  }
}
```

Collection Update JSON:

Copyright @2024 ZBlox. All Rights Reserved

json

```
{
 "type": "collection_update",
 "collectionId": "users-collection",
 "operation": "add",
 "item": {
  "type": "User",
  "properties": {
   "name": "",
   "email": "",
   "active": true
  }
 }
}
```

5. Tables and Data Grids

Basic Table Structure

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Simple static table
table {
 thead {
  row {
   header { "Name" }
   header { "Email" }
   header { "Status" }
   header { "Actions" }
  }
 }
 tbody {
  row {
   cell { "John Doe" }
   cell { "john@example.com" }
   cell { badge("Active") { color: "green" } }
   cell {
    button("Edit") { onclick: editUser("john") }
    button("Delete") { onclick: deleteUser("john") }
   }
  }
  row {
   cell { "Jane Smith" }
   cell { "jane@example.com" }
   cell { badge("Inactive") { color: "red" } }
   cell {
    button("Edit") { onclick: editUser("jane") }
    button("Delete") { onclick: deleteUser("jane") }
   }
  }
 }
}
```

Dynamic Tables with Server Binding

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Data-bound table with server-side collection
table(bind: users) {
 columns {
  column("name") {
   header: "Full Name"
   sortable: true
   render: user => text { user.firstName + " " + user.lastName }
  }

  column("email") {
   header: "Email Address"
   sortable: true
   searchable: true
  }

  column("department") {
   header: "Department"
   filterable: true
   render: user => text { user.department.name }
  }

  column("status") {
   header: "Status"
   render: user => badge(user.status) {
    color: user.status === "active" ? "green" : "red"
   }
  }

  column("actions") {
   header: "Actions"
   render: user => actions {
    button("Edit") { onclick: editUser(user) }
    button("Delete") { onclick: deleteUser(user), confirm: "Are you sure?" }
   }
  }
 }

 // Table features
 pagination: true
 pageSize: 25
 search: true
 onRowClick: user => viewUserDetails(user)
}
```

Copyright @2024 ZBlox. All Rights Reserved

Editable Data Grid

```javascript
// Spreadsheet-style editable table
dataGrid(bind: salesData) {
 columns {
  column("product") { header: "Product", editable: false }
  column("q1Sales") { header: "Q1 Sales", editable: true, type: "number" }
  column("q2Sales") { header: "Q2 Sales", editable: true, type: "number" }
  column("q3Sales") { header: "Q3 Sales", editable: true, type: "number" }
  column("q4Sales") { header: "Q4 Sales", editable: true, type: "number" }
  column("total") {
   header: "Total"
   editable: false
   computed: row => row.q1Sales + row.q2Sales + row.q3Sales + row.q4Sales
  }
 }

 onCellChange: (row, column, newValue) => updateSalesData(row.id, column, newValue)
 onRowAdd: () => addNewSalesRow()
 onRowDelete: (row) => deleteSalesRow(row.id)
}
```

6. Documents and Rich Content

Document Structure

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Document layout
document {
 title { "Quarterly Report Q4 2024" }

 abstract {
  paragraph { "This document summarizes the key findings and metrics for Q4 2024..." }
 }

 section("Executive Summary") {
  paragraph { "Our company achieved significant growth in Q4 2024..." }

  highlights {
   bullet { "Revenue increased by 25%" }
   bullet { "Customer acquisition up 40%" }
   bullet { "New market expansion in 3 regions" }
  }
 }

 section("Financial Performance") {
  subsection("Revenue Analysis") {
   paragraph { "Total revenue for Q4 was $2.3M, representing..." }

   chart {
    type: "line"
    data: revenueData
    title: "Revenue Trend"
   }
  }

  subsection("Cost Analysis") {
   table(bind: costBreakdown) {
    columns {
     column("category") { header: "Cost Category" }
     column("amount") { header: "Amount", format: "currency" }
     column("percentage") { header: "% of Total", format: "percent" }
    }
   }
  }
 }

 appendix("Data Sources") {
  list {
   item { link("Financial System", href: "/reports/financial") }
```

Copyright @2024 ZBlox. All Rights Reserved

```
      item { link("CRM Data", href: "/reports/crm") }
      item { link("Analytics Dashboard", href: "/analytics") }
    }
  }
}
```

Rich Text and Media

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Rich content page
article {
 header {
  title { "Getting Started Guide" }
  metadata {
   author { "Documentation Team" }
   publishDate { "2024-12-15" }
   tags { "tutorial", "getting-started", "basics" }
  }
 }

 content {
  paragraph {
   text { "Welcome to our platform! This guide will help you " }
   emphasis { "quickly get started" }
   text { " with the basic features." }
  }

  image {
   src: "/images/dashboard-overview.png"
   alt: "Dashboard Overview Screenshot"
   caption: "The main dashboard interface"
  }

  codeBlock(language: "javascript") {
   '''
   // Example template code
   dashboard {
    header { "My Dashboard" }
    content { "Welcome!" }
   }
   '''
  }

  callout(type: "warning") {
   paragraph { "Make sure to save your work regularly to avoid data loss." }
  }

  video {
   src: "/videos/tutorial-intro.mp4"
   poster: "/images/video-thumbnail.jpg"
   controls: true
  }
```

Copyright @2024 ZBlox. All Rights Reserved

```
  }
}
```

7. Hyperlinks as Function Calls

Basic Navigation Links

```javascript
// Traditional href-style links
navigation {
  link("Home") { href: "/dashboard" }
  link("Users") { href: "/users" }
  link("Reports") { href: "/reports" }
}

// Function call links - these become service function calls
navigation {
  link("Dashboard") { onclick: navigateTo("dashboard") }
  link("User Profile") { onclick: showUserProfile(currentUser.id) }
  link("Settings") { onclick: openSettings() }
}
```

Function Call Translation to JSON

When a function call like `navigateTo("dashboard")` is triggered:

Template Definition:

```javascript
button("Go to Dashboard") { onclick: navigateTo("dashboard") }
```

Compiles to HTMLNode:

```javascript
HTMLNode.element("button", {
  onclick: FunctionCall("navigateTo", ["dashboard"])
}, [HTMLNode.text("Go to Dashboard")])
```

Runtime JSON Message:

Copyright @2024 ZBlox. All Rights Reserved

```json
{
 "type": "service_call",
 "function": "navigateTo",
 "parameters": ["dashboard"],
 "context": {
  "sessionId": "user-session-123",
  "timestamp": "2024-12-15T10:30:00Z",
  "sourceElement": "button-dashboard-nav"
 }
}
```

Complex Function Calls with Object Parameters

```javascript
// Function call with object parameter
button("Update User") {
 onclick: updateUser({
  id: user.id,
  name: user.name,
  email: user.email,
  preferences: user.preferences
 })
}

// Function call with multiple parameters
link("Send Message") {
 onclick: sendMessage(recipient.id, messageText, attachments, priority: "high")
}
```

JSON Message for Complex Call:

Copyright @2024 ZBlox. All Rights Reserved

json

```
{
 "type": "service_call",
 "function": "updateUser",
 "parameters": [{
  "id": 123,
  "name": "John Doe",
  "email": "john@example.com",
  "preferences": {
   "theme": "dark",
   "notifications": true
  }
 }],
 "context": {
  "sessionId": "user-session-123",
  "timestamp": "2024-12-15T10:30:00Z",
  "sourceElement": "button-update-user"
 }
}
```

8. Advanced Layout Patterns

Header and Positioning

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Application shell with fixed header
app {
 header(position: "fixed", top: 0) {
  navbar {
   brand { "My Application" }
   navigation {
    link("Dashboard") { onclick: navigateTo("dashboard") }
    link("Users") { onclick: navigateTo("users") }
   }
   userMenu {
    dropdown {
     trigger { avatar(src: currentUser.avatar) }
     menu {
      item("Profile") { onclick: showProfile() }
      item("Settings") { onclick: showSettings() }
      divider
      item("Logout") { onclick: logout() }
     }
    }
   }
  }
 }

 main(marginTop: "60px") { // Account for fixed header
  sidebar(position: "fixed", left: 0, width: "250px") {
   navigation {
    section("Main") {
     navItem("Dashboard") { onclick: navigateTo("dashboard") }
     navItem("Analytics") { onclick: navigateTo("analytics") }
    }
    section("Management") {
     navItem("Users") { onclick: navigateTo("users") }
     navItem("Reports") { onclick: navigateTo("reports") }
    }
   }
  }

  content(marginLeft: "250px", padding: "20px") {
   // Main content area
   slot("content") // Content injected here
  }
 }
```

Copyright @2024 ZBlox. All Rights Reserved

```
  footer(position: "fixed", bottom: 0) {
    text { "© 2025 My Company. All rights reserved." }
  }
}
```

Responsive Layout

javascript

```
// Responsive grid that adapts to screen size
responsiveGrid {
  breakpoints: {
    mobile: "768px"
    tablet: "1024px"
    desktop: "1200px"
  }

  columns: {
    mobile: 1
    tablet: 2
    desktop: 3
  }

  for (item in items) {
    gridItem {
      card {
        image { src: item.image }
        title { item.name }
        description { item.description }
        actions {
          button("View") { onclick: viewItem(item) }
          button("Edit") { onclick: editItem(item) }
        }
      }
    }
  }
}
```

Modal and Overlay Patterns

Copyright @2024 ZBlox. All Rights Reserved

javascript

```
// Modal dialog
modal(bind: showUserDialog) {
 header {
  title { "Edit User" }
  closeButton { onclick: closeUserDialog() }
 }

 body {
  form(bind: selectedUser) {
   field {
    label { "Name" }
    input(bind: selectedUser.name)
   }
   field {
    label { "Email" }
    input(bind: selectedUser.email)
   }
  }
 }

 footer {
  button("Save") {
   onclick: saveUser(selectedUser)
   type: "primary"
  }
  button("Cancel") {
   onclick: closeUserDialog()
   type: "secondary"
  }
 }
}

// Slide-out panel
slidePanel(bind: showNotifications, position: "right") {
 header { "Notifications" }

 content {
  for (notification in notifications) {
   notificationItem {
    icon { notification.type }
    message { notification.message }
    timestamp { notification.createdAt }
    if (notification.unread) {
```

Copyright @2024 ZBlox. All Rights Reserved

```
        markAsRead { onclick: markNotificationRead(notification) }
      }
     }
    }
   }
  }
```

This tutorial demonstrates how the declarative template syntax compiles to HTMLNode trees, integrates with server-side backing objects through two-way data binding, and translates function calls to JSON messages for service communication. The system provides a clean separation between presentation logic and business logic while maintaining real-time synchronization between client and server state.

Copyright @2024 ZBlox. All Rights Reserved