

# QUICK-START

## Quick Start Guide

### Get Up and Running in 15 Minutes

#### Step 1: Install Dependencies (2 minutes)

```
cd prototypes/scorm  
npm install
```

This installs the build tools needed to create SCORM packages.

#### Step 2: Explore a Template (3 minutes)

Open `templates/drag-drop-template.html` in your browser:

*# Mac:*

```
open templates/drag-drop-template.html
```

*# Windows:*

```
start templates/drag-drop-template.html
```

*# Linux:*

```
xdg-open templates/drag-drop-template.html
```

**Try the interaction:** - Drag organelles into categories - Click “Check Answers” - Click “Reset” to try again - Click “Hint” for help

#### Step 3: Customize with Claude Code (5 minutes)

Ask Claude Code to modify the template:

```
"Change the drag-drop-template.html to be about US state capitals  
instead of cell organelles.  
Create 6 state-capital pairs. Change the colors to red, white, and  
blue."
```

Claude will modify the HTML for you. Refresh your browser to see changes.

## Step 4: Test SCORM API (2 minutes)

Open `testing/scorm-validator.html` in your browser:

open `testing/scorm-validator.html`

**Test the mock SCORM API:** 1. Click “Initialize” 2. Click “Test Completion” 3. Click “Test Score (85)” 4. Watch the log to see SCORM calls

## Step 5: Create Your First Module (3 minutes)

### 1. Copy a template:

```
mkdir example-module
cp templates/drag-drop-template.html example-module/index.html
cp templates/scorm-wrapper.js example-module/
```

### 2. Build SCORM package:

```
npm run build
```

### 3. Follow the prompts:

- Source folder: `example-module`
- Title: “My First Learning Object”
- Description: “Testing SCORM package creation”
- Version: 1.0
- Output filename: “my-first-module”

4. **Result:** You’ll find `dist/my-first-module.zip` ready to upload to your LMS!

---

## What to Do Next

### For Learning Designers:

1. **Read the main README** (`README.md`)
  - Understand the full workflow
  - See all available templates
  - Learn customization options
2. **Review Higher Ed strategy** (`docs/HIGHER-ED-STRATEGY.md`)
  - Understand the business case
  - See ROI calculations
  - Learn scaling strategies
3. **Start building:**
  - Choose a course that needs interactive content
  - Pick appropriate template
  - Work with Claude Code to customize

## For IT Staff:

1. **Complete the IT questionnaire** (docs/IT-REQUIREMENTS.md)
  - o Gather LMS specifications
  - o Document any restrictions
  - o Share with Learning Team
2. **Review testing checklist** (testing/test-checklist.md)
  - o Understand validation process
  - o Set up testing workflow
  - o Establish sign-off procedure
3. **Read troubleshooting guide** (docs/TROUBLESHOOTING.md)
  - o Learn common issues
  - o Know how to diagnose problems
  - o Have solutions ready

## For Administrators:

1. **Review business case** (docs/HIGHER-ED-STRATEGY.md)
    - o See 3-year ROI: \$50K-\$75K
    - o Understand competitive advantages
    - o Plan pilot program
  2. **Allocate resources:**
    - o 1 Learning Technologist (40% FTE) - Year 1
    - o 1 Instructional Designer (20% FTE) - Year 1
    - o Budget: \$60K-\$70K Year 1, \$30K-\$35K ongoing
  3. **Set timeline:**
    - o Months 1-3: Build foundation
    - o Months 4-6: Pilot
    - o Months 7-12: Scale
- 

## Common First Questions

**Q: Do I need to know how to code?** A: No! Learning designers work with Claude Code to make changes. Claude handles the coding.

**Q: Will this work with our LMS?** A: Yes, if your LMS supports SCORM 1.2 (99% do). Have IT complete the questionnaire to confirm.

**Q: How long does it take to create a learning object?** A: First one: 2-3 hours. After building your library: 30-60 minutes.

**Q: What if something breaks?** A: Check docs/TROUBLESHOOTING.md for solutions to common issues.

**Q: Can we customize the look and feel?** A: Absolutely! You have full control over colors, fonts, layout, everything.

**Q: Is this accessible?** A: Yes, templates are built to WCAG 2.1 AA standards. See docs/ACCESSIBILITY-GUIDE.md.

**Q: How big are the packages?** A: Typical interactive module: 2-10 MB. Well within LMS limits.

---

## File Structure at a Glance

```
scorm/
├── README.md                ← Start here!
├── QUICK-START.md          ← You are here
├── package.json            ← Node dependencies
├── templates/              ← Ready-to-use components
│   ├── drag-drop-template.html
│   ├── scorm-wrapper.js
│   └── ...
├── docs/                  ← Comprehensive guides
│   ├── IT-REQUIREMENTS.md
│   ├── HIGHER-ED-STRATEGY.md
│   ├── TROUBLESHOOTING.md
│   └── ...
├── testing/               ← Testing tools
│   ├── scorm-validator.html
│   ├── test-checklist.md
│   └── ...
├── scripts/               ← Build automation
│   └── build-scorm.js
└── dist/                  ← Generated SCORM packages
    └── *.zip
```

---

## Getting Help

### In this folder:

- README.md - Complete documentation
- docs/TROUBLESHOOTING.md - Problem solving
- testing/test-checklist.md - Validation steps

### With Claude Code:

Ask Claude to help with customization: - “Make the buttons bigger” - “Change colors to match our branding” - “Add 3 more questions” - “Fix this error message”

## Your team:

- IT Team: Technical issues, LMS problems
  - Learning Design Team: Instructional design, content
  - Administrators: Budget, resources, strategy
- 

## Success Checklist

After completing this quick start, you should have:

- Installed Node dependencies (`npm install`)
- Opened and interacted with a template
- Seen the SCORM validator in action
- Built your first SCORM package
- Located the ZIP file in `dist/` folder

**Next milestone:** Upload your package to LMS staging environment!

---

## Tips for Success

1. **Start small** - Pick one high-value course for pilot
  2. **Test early** - Use SCORM Cloud before LMS testing
  3. **Document everything** - Keep notes on what works
  4. **Iterate quickly** - Get feedback from real learners
  5. **Build library** - Create reusable components
  6. **Measure impact** - Track engagement and outcomes
- 

**Ready to build something amazing?** 🚀

Open `README.md` for full documentation, or dive right in with the templates!