Tutorial (Web) Track Selector

Track Selector

Build the track selector

Many input file formats that alphaTab supports can contain more than just one instrument. There can be vocals, guitars, drums or other instruments where the notation is available, and users would like to switch between the individual tracks.

The track selector will be put in the sidebar, so let's extend our `index.html` with a template that we will use for all items in the track selector and some place in the sidebar where the items will go.

New DOM New CSS

```html
<!-- New in head tag -->
<script src="https://kit.fontawesome.com/b43f0e512e.js"></script>
<!-- Changed sidebar -->
<div class="at-sidebar">
  <div class="at-sidebar-content">
    <div class="at-track-list"></div>
  </div>
</div>
<!-- New below wrapper -->
<template id="at-track-template">
  <div class="at-track">
    <div class="at-track-icon">
      <i class="fas fa-guitar"></i>
    </div>
    <div class="at-track-details">
      <div class="at-track-name"></div>
    </div>
  </div>
</template>
```

Let's take a closer look at the new UI element. We have one column for the icon of the track and another one for the details of the track. In the details we still only show the name of the track. Other controls for volume and muting will be added later. In this tutorial, we use Font Awesome to display some icons.

> ⚠ **CAUTION**
>
> Do not use this Font Awesome CDN link from this tutorial. Head over to https://fontawesome.com/ to create your own personal kit for usage on your website.

Now the real magic happens as we will hook this up with alphaTab. We will once again listen to some events and fill the UI accordingly. This will also be the first time we use UI interaction to tell alphaTab to do something. When a user clicks on an item, we want to render the selected track.

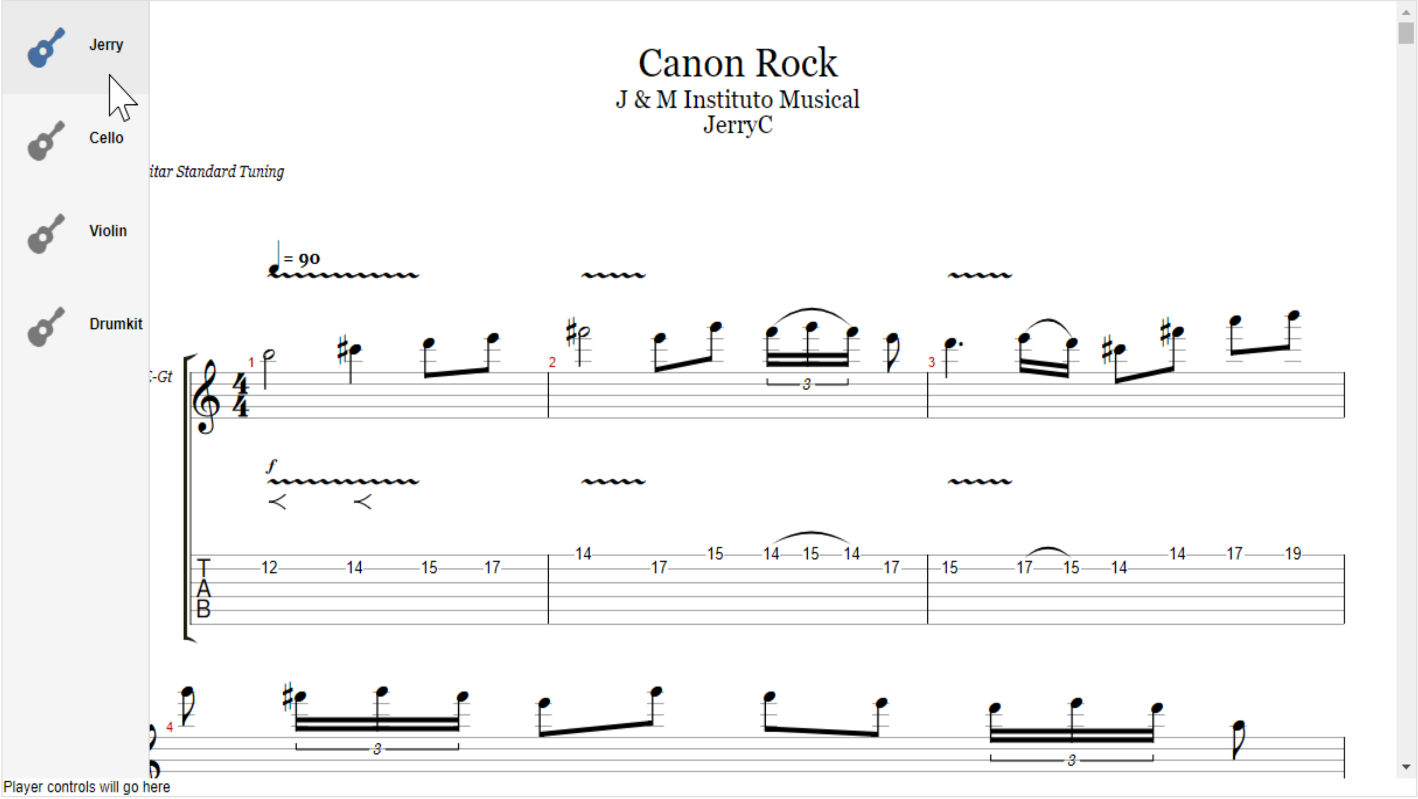
```
// helper function to create individual items
function createTrackItem(track) {
  const trackItem = document
    .querySelector("#at-track-template")
    .content.cloneNode(true).firstElementChild;
  trackItem.querySelector(".at-track-name").innerText = track.name;
  trackItem.track = track;
  trackItem.onclick = (e) => {
    e.stopPropagation();
    // here we use some API function of alphaTab.
    // check the reference docs for the details.
    api.renderTracks([track]);
  };
  return trackItem;
}


const trackList = wrapper.querySelector(".at-track-list");
// fill track list when the score is loaded
api.scoreLoaded.on((score) => {
  // clear items
  trackList.innerHTML = "";
  // generate a track item for all tracks of the score
  score.tracks.forEach((track) => {
    trackList.appendChild(createTrackItem(track));
  });
});

// mark the rendered track as active in the list
api.renderStarted.on(() => {
  // collect tracks being rendered
  const tracks = new Map();
  // here we access the currently rendered tracks of alphaTab
  api.tracks.forEach((t) => {
    tracks.set(t.index, t);
  });
  // mark the item as active or not
  const trackItems = trackList.querySelectorAll(".at-track");
  trackItems.forEach((trackItem) => {
    if (tracks.has(trackItem.track.index)) {
      trackItem.classList.add("active");
    } else {
      trackItem.classList.remove("active");
    }
  });
});
```

The code above should be easy understandable but let's have a closer look at what happens here.

We have a small helper function which creates the DOM element for a given track. It uses the `<template>` element from before to create this element, and then we just fill in the name. Upon click we tell alphaTab to render the given track.

Below that helper function, we listen to the `scoreLoaded` event which is fired once the data model for the song is loaded. There we iterate through each track and create an item for the track selector and place it in the sidebar.

Last but not least, we add another listener to the `renderStarted` event that marks the currently rendered tracks as active in the sidebar with a CSS class.



Final File

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>AlphaTab Tutorial</title>
    <script
src="https://cdn.jsdelivr.net/npm/@coderline/alphatab@latest/dist/alphaTab.j
</script>
    <script src="https://kit.fontawesome.com/b43f0e512e.js"></script>
    <style type="text/css">
      body {
        font-family: Arial, Helvetica, sans-serif;
        font-size: 12px;
      }
      .at-wrap {
        width: 80vw;
        height: 80vh;
        margin: 0 auto;
        border: 1px solid rgba(0, 0, 0, 0.12);
        display: flex;
```

```
          flex-direction: column;
          overflow: hidden;
          position: relative;
        }
        .at-content {
          position: relative;
          overflow: hidden;
          flex: 1 1 auto;
        }
        /** Sidebar **/
        .at-sidebar {
          position: absolute;
          top: 0;
          left: 0;
          bottom: 0;
          max-width: 70px;
          width: auto;
          display: flex;
          align-content: stretch;
          z-index: 1001;
          overflow: hidden;
          border-right: 1px solid rgba(0, 0, 0, 0.12);
          background: #f7f7f7;
        }
        .at-sidebar:hover {
          max-width: 400px;
          transition: max-width 0.2s;
          overflow-y: auto;
        }

        .at-viewport {
          overflow-y: auto;
          position: absolute;
          top: 0;
          left: 70px;
          right: 0;
          bottom: 0;
          padding-right: 20px;
        }
        .at-footer {
          flex: 0 0 auto;
          background: #436d9d;
          color: #fff;
        }
```

```css
/** Overlay **/

.at-overlay {
  /** Fill Parent */
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  z-index: 1002;

  /* Blurry dark shade */
  backdrop-filter: blur(3px);
  background: rgba(0, 0, 0, 0.5);

  /** center content */
  display: flex;
  justify-content: center;
  align-items: flex-start;
}

.at-overlay-content {
  /* white box with drop-shadow */
  margin-top: 20px;
  background: #fff;
  box-shadow: 0px 5px 10px 0px rgba(0, 0, 0, 0.3);
  padding: 10px;
}

/** Track selector **/
.at-track {
  display: flex;
  position: relative;
  padding: 5px;
  transition: background 0.2s;
  cursor: pointer;
}
.at-track:hover {
  background: rgba(0, 0, 0, 0.1);
}
.at-track > .at-track-icon,
.at-track > .at-track-details {
  display: flex;
```

```
        flex-direction: column;
        justify-content: center;
      }
      .at-track > .at-track-icon {
        flex-shrink: 0;
        font-size: 32px;
        opacity: 0.5;
        transition: opacity 0.2s;
        width: 64px;
        height: 64px;
        margin-right: 5px;
        align-items: center;
      }
      .at-track-name {
        font-weight: bold;
        margin-bottom: 5px;
      }
      .at-track:hover > .at-track-icon {
        opacity: 0.8;
      }
      .at-track.active {
        background: rgba(0, 0, 0, 0.03);
      }
      .at-track.active > .at-track-icon {
        color: #4972a1;
        opacity: 1;
      }
      .at-track > .at-track-name {
        font-weight: 500;
      }
    </style>
  </head>
  <body>
    <div class="at-wrap">
      <div class="at-overlay">
        <div class="at-overlay-content">
          Music sheet is loading
        </div>
      </div>
      <div class="at-content">
        <div class="at-sidebar">
          <div class="at-sidebar-content">
            <div class="at-track-list"></div>
          </div>
    </div>
    <div class="at-viewport">
      <div class="at-main"></div>
    </div>
  </div>
  <div class="at-controls">
    Player controls will go here
  </div>
</div>

<template id="at-track-template">
  <div class="at-track">
    <div class="at-track-icon">
      <i class="fas fa-guitar"></i>
    </div>
    <div class="at-track-details">
      <div class="at-track-name"></div>
    </div>
  </div>
</template>

<script type="text/javascript">
  // load elements
  const wrapper = document.querySelector(".at-wrap");
  const main = wrapper.querySelector(".at-main");

  // initialize alphatab
  const settings = {
    file: "https://www2.alphatab.net/files/canon.gp",
  };
  const api = new alphaTab.AlphaTabApi(main, settings);

  // overlay logic
  const overlay = wrapper.querySelector(".at-overlay");
  api.renderStarted.on(() => {
    overlay.style.display = "flex";
  });
  api.renderFinished.on(() => {
    overlay.style.display = "none";
  });

  // track selector
  function createTrackItem(track) {
    const trackItem = document
```

```
          .querySelector("#at-track-template")
          .content.cloneNode(true).firstElementChild;
        trackItem.querySelector(".at-track-name").innerText = track.name;
        trackItem.track = track;
        trackItem.onclick = (e) => {
          e.stopPropagation();
          api.renderTracks([track]);
        };
        return trackItem;
      }
      const trackList = wrapper.querySelector(".at-track-list");
      api.scoreLoaded.on((score) => {
        // clear items
        trackList.innerHTML = "";
        // generate a track item for all tracks of the score
        score.tracks.forEach((track) => {
          trackList.appendChild(createTrackItem(track));
        });
      });
      api.renderStarted.on(() => {
        // collect tracks being rendered
        const tracks = new Map();
        api.tracks.forEach((t) => {
          tracks.set(t.index, t);
        });
        // mark the item as active or not
        const trackItems = trackList.querySelectorAll(".at-track");
        trackItems.forEach((trackItem) => {
          if (tracks.has(trackItem.track.index)) {
            trackItem.classList.add("active");
          } else {
            trackItem.classList.remove("active");
          }
        });
      });
    </script>
  </body>
</html>
```