

Transition Flow: LocalStorage Backend API (for WorkNest Canvas Layouts)

Current Setup (Frontend)

- HTML5 Canvas using 2D context
- Layout data tracked in a JavaScript array (layoutObjects)
- Layouts saved/retrieved per office using localStorage
- Office ID extracted from URL: ?office=office-123

Goal

Replace localStorage with REST API calls to a backend that:

- Saves layout JSON per office
- Retrieves layout JSON by office ID
- Optionally stores metadata (e.g., last modified, user ID)

1. Layout Object Format (unchanged)

Each canvas element is represented like this:

```
{  
  id: "desk-1",  
  type: "desk",  
  x: 100,  
  y: 150,  
  width: 60,  
  height: 40  
}
```

All elements are tracked in:

```
const layoutObjects = [ /* array of layout elements */ ];
```

2. Replace localStorage Saving with API Call

Before (localStorage):

```
localStorage.setItem("worknest_layouts", JSON.stringify(allLayouts));
```

After (Backend API):

```
function saveLayoutToBackend() {  
  fetch("/api/save-layout", {  
    method: "POST",  
    headers: { "Content-Type": "application/json" },  
    body: JSON.stringify({  
      officeId: getOfficeIdFromUrl(),  
      layout: layoutObjects,  
      lastModified: new Date().toISOString()  
    })  
  });  
}
```

3. Replace localStorage Loading with API Call

Before:

```
const layouts = JSON.parse(localStorage.getItem("worknest_layouts")) || {};  
const layout = layouts[officeId];
```

After:

```
function loadLayoutFromBackend() {  
  
  fetch(`/api/get-layout?officeId=${getOfficeIdFromUrl()}`)  
  
    .then(res => res.json())  
  
    .then(data => {  
  
      if (data?.layout) {  
  
        redrawFromLayout(data.layout);  
  
      } else {  
  
        initBlankCanvas();  
  
      }  
  
    });  
  
}
```

4. API Contract for Backend Developer

POST /api/save-layout

Request Body:

```
{  
  
  "officeId": "office-123",  
  
  "layout": [/* array of layout elements */],  
  
  "lastModified": "2025-06-04T12:00:00.000Z"  
  
}
```

Response:

```
{ "success": true, "message": "Layout saved" }
```

GET /api/get-layout?officeId=office-123

Response:

```
{
  "officeId": "office-123",
  "layout": [/* array of layout elements */],
  "lastModified": "2025-06-04T12:00:00.000Z"
}
```

5. Functions to Keep

```
function getOfficeIdFromUrl() {
  const params = new URLSearchParams(window.location.search);
  return params.get("office") || "default-office";
}
```

Summary of Changes

Feature	Before (localStorage)	After (Backend API)
Save layout	localStorage.setItem()	fetch('/api/save-layout', ...)
Load layout	localStorage.getItem()	fetch('/api/get-layout?officeId=...')
Storage format	JSON (per office in browser)	JSON (per office in DB)
Auth/scoping	None (browser-local only)	Can add auth tokens, sessions

Extras (Optional for Backend)

- Use MongoDB or PostgreSQL to store layout per office ID
- Add authentication (user ID/session)
- Add timestamps, versions, backups
- Enable layout sharing / permissions in the future

