

# Authorization Fix Summary

---

## Problem

Free users were getting a **403 Forbidden error** when trying to access the Integration Hub because the system tried to create a project on hub access, which was blocked by subscription limits (maxProjects: 0 for free users).

## Solution

Implemented a “try before you buy” model where free users can access and test all hub features but cannot save projects or download code.

---

## Changes Made

### 1. Backend: Subscription Policy Service ( `server/services/subscriptionPolicyService.ts` )

#### Changed Free Tier Limits

```
free: {
  maxProjects: 1, // Changed from 0 to 1 (allows 1 temporary preview project)
  retentionDays: 0, // No retention - cleaned up on logout
  maxDownloads: 0, // No downloads
  downloadPeriod: 'monthly',
  teamSize: 1,
  canDownload: false,
}
```

#### Added Project Save Check Method

```
static canSaveProject(subscriptionStatus: string): {
  allowed: boolean;
  message?: string;
}
```

This method checks if a user can permanently save projects. Free users are blocked with a friendly upgrade message.

#### Updated Authorization Messages

- Improved error messages for free users when they hit project limits
  - Added context-aware messages that explain the preview/trial model
-

## 2. Backend: API Routes ( `server/routes.ts` )

### Added Explicit Save Endpoint

```
POST /api/projects/:id/save
```

- Requires authentication
- Checks `SubscriptionPolicyService.canSaveProject()`
- Returns 403 with `requiresUpgrade: true` flag for free users
- Shows upgrade prompt on frontend

### Enhanced Download Endpoints

Updated all download endpoints to include the `requiresUpgrade` flag:

- `/api/projects/:id/download/xslt`
- `/api/projects/:id/download/dataweave`
- `/api/projects/:id/download/mapping-file`
- `/api/projects/:id/download/mapping-document`

```
return res.status(403).json({
  message: downloadCheck.message,
  remaining: downloadCheck.remaining || 0,
  requiresUpgrade: req.user.subscriptionStatus === 'free'
});
```

## 3. Frontend: Integration Hub Page ( `client/src/pages/integration-hub.tsx` )

### Added Save Project Functionality

```
const saveProjectMutation = useMutation({
  mutationFn: async () => {
    const response = await apiRequest(`/api/projects/${currentProject?.id}/save`, "POST");
    // ... handle response
  },
  onError: (error) => {
    if (error.message.includes('upgrade') || error.message.includes('paid plan')) {
      if (confirm(`${error.message}\n\nWould you like to view our pricing plans?`)) {
        window.location.href = '/pricing';
      }
    }
  },
});
```

## Wired Up Save Button

```
<Button
  variant="outline"
  size="sm"
  onClick={handleSaveProject}
  disabled={saveProjectMutation.isPending}
>
  {saveProjectMutation.isPending ? 'Saving...' : 'Save'}
</Button>
```

## 4. Frontend: Validation Success Component ( `client/src/components/validation-success.tsx` )

### Added Free User Detection

```
const isFreeUser = user?.subscriptionStatus === 'free';
```

### Updated Download Logic

- Check for both `isFreeUser` and legacy `isTrial` status
- Show upgrade dialog immediately for free users
- Disable download buttons with lock icon

### Enhanced UI Messages

```
{(isFreeUser || isTrial) && (
  <Alert>
    <Crown className="h-4 w-4" />
    <AlertDescription>
      <p>Downloads are disabled for Free users</p>
      <p>You can preview and test all features. Upgrade to download files</p>
      <Button asChild>
        <Link href="/pricing">Upgrade Now</Link>
      </Button>
    </AlertDescription>
  </Alert>
)}
```

## How It Works Now

### For FREE Users:

#### CAN DO:

- Access the Integration Hub
- Upload source and target files
- Use AI-powered field mapping
- Generate XSLT and DataWeave transformations
- Preview all generated code
- Run validation tests (5 rows preview)
- Test all features in the workflow

### **CANNOT DO:**

- Save projects permanently (shows upgrade prompt)
- Download generated code files (shows upgrade prompt)
- Create multiple projects (shows upgrade prompt after 1st project)

### **For PAID Users (One-Time, Monthly, Annual):**

#### **CAN DO:**

- Everything free users can do
  - Save projects permanently
  - Download all generated files (XSLT, DataWeave, mappings, docs)
  - Create multiple projects based on their tier
  - Keep projects for retention period based on their tier
- 

## User Experience Flow

1. **Free user registers and logs in**
  2. **Automatically redirected to Integration Hub**
  3. **System creates 1 temporary preview project** (no more 403 error!)
  4. **User uploads files and uses all features freely**
  5. **When trying to save:**
    - Shows friendly dialog: "Free users can preview and test the hub, but cannot save projects permanently. Upgrade to a paid plan to save your work."
    - Offers link to pricing page
  6. **When trying to download:**
    - Shows lock icon on download buttons
    - Shows banner: "Downloads are disabled for Free users. You can preview and test all features. Upgrade to download files"
    - Offers "Upgrade Now" button
- 

## Testing Recommendations

### **Test Case 1: Free User Access**

1. Register as a new free user
2. Navigate to /hub
3. **Expected:** Hub loads successfully with 1 preview project created
4. Upload source and target files
5. Generate field mappings
6. Generate XSLT/DataWeave code
7. **Expected:** All features work, code is visible in preview

### **Test Case 2: Free User Save Attempt**

1. As a free user in the hub
2. Click "Save" button
3. **Expected:** Dialog appears with upgrade message and link to pricing

## Test Case 3: Free User Download Attempt

1. As a free user, complete the workflow to validation step
2. Try to click any download button
3. **Expected:** Buttons are disabled with lock icon
4. **Expected:** Banner shows upgrade message with “Upgrade Now” button

## Test Case 4: Paid User Full Access

1. Register/upgrade to paid tier
2. Navigate to /hub
3. Complete full workflow
4. **Expected:** Save button works normally
5. **Expected:** All download buttons work and files download successfully

## Test Case 5: Multiple Projects

1. As a free user with 1 existing project
  2. Try to create a new project (click “New Project”)
  3. **Expected:** Error message: “You already have an active preview project. Upgrade to a paid plan to create and save multiple projects.”
- 

## Files Modified

1. server/services/subscriptionPolicyService.ts - Updated limits and added save check
  2. server/routes.ts - Added save endpoint and enhanced download responses
  3. client/src/pages/integration-hub.tsx - Added save functionality with upgrade prompts
  4. client/src/components/validation-success.tsx - Updated to check free user status and show upgrade prompts
- 

## Git Commit

Changes have been committed locally:

```
commit 77255f7
Fix: Allow free users to access Integration Hub in preview mode
```

**Note:** The push to GitHub failed due to authentication issues. You can push manually with:

```
cd /home/ubuntu/integration-hub
git push origin main-backup
```

Or configure SSH/token authentication and retry.

---

## Production Deployment Checklist

---

Before deploying to production:

1.  Ensure PostgreSQL database is running and accessible
  2.  Set DATABASE\_URL environment variable
  3.  Set SESSION\_SECRET environment variable
  4.  Run database migrations if any
  5.  Test with actual free user registration
  6.  Verify Stripe webhook integration for subscription upgrades
  7.  Test upgrade flow from free to paid
  8.  Monitor error logs for any 403/auth issues
- 

## Success Metrics

---

After deployment, monitor:

- **No more 403 errors for free users accessing /hub**
  - Increase in free user engagement (file uploads, mappings generated)
  - Conversion rate from free to paid (upgrade clicks from prompts)
  - Reduction in support tickets about “can’t access hub”
- 

## Notes

---

- Free users’ preview projects are ephemeral (retentionDays: 0)
- Projects are cleaned up when free users log out or periodically
- This implements a true “freemium” model where users can try before buying
- All upgrade prompts link to `/pricing` page for seamless conversion