

MailerLite Custom Fields Fix - Automatic Field Creation

Problem Identified

Issue: Emails were being added to MailerLite, but custom fields were not being populated.

Root Cause: MailerLite requires custom fields to be **created first** before they can be used. The previous code was trying to send field values without ensuring the fields existed, causing MailerLite to silently ignore them.

Solution Implemented

Updated `/api/mailerlite.js` to **automatically create custom fields** if they don't exist.

How It Works:

1. **Check Existing Fields** - Before adding a subscriber, the function fetches all existing custom fields from MailerLite
 2. **Create Missing Fields** - Any required fields that don't exist are automatically created
 3. **Add/Update Subscriber** - Once fields are ready, the subscriber is added with all custom field values
-

Required Custom Fields

The function now ensures these 9 custom fields exist:

Field Key	Field Name	Type	Example Value
archetype	Archetype	TEXT	"The Sage"
archetype_percent	Archetype Percent	NUMBER	32
test_date	Test Date	TEXT	"1/14/2026"
archetype_1	Archetype 1	TEXT	"The Sage"
archetype_1_percent	Archetype 1 Percent	NUMBER	32
archetype_2	Archetype 2	TEXT	"The Visionary"
archetype_2_percent	Archetype 2 Percent	NUMBER	24
archetype_3	Archetype 3	TEXT	"The Builder"
archetype_3_percent	Archetype 3 Percent	NUMBER	18

Note: The `name` field is a built-in MailerLite field and doesn't need to be created.

Code Changes

New Function: `ensureCustomFieldsExist()`

This function:

1. Fetches existing fields from MailerLite API: `GET /api/fields`
2. Compares with required fields list
3. Creates any missing fields: `POST /api/fields`
4. Logs all actions for debugging

Updated Main Handler:

```
export default async function handler(req, res) {
  // Step 1: Ensure fields exist
  await ensureCustomFieldsExist();

  // Step 2: Add/update subscriber
  // ... rest of the code
}
```

Field Definition:

```
const REQUIRED_FIELDS = [
  { key: 'archetype', name: 'Archetype', type: 'TEXT' },
  { key: 'archetype_percent', name: 'Archetype Percent', type: 'NUMBER' },
  { key: 'test_date', name: 'Test Date', type: 'TEXT' },
  { key: 'archetype_1', name: 'Archetype 1', type: 'TEXT' },
  { key: 'archetype_1_percent', name: 'Archetype 1 Percent', type: 'NUMBER' },
  { key: 'archetype_2', name: 'Archetype 2', type: 'TEXT' },
  { key: 'archetype_2_percent', name: 'Archetype 2 Percent', type: 'NUMBER' },
  { key: 'archetype_3', name: 'Archetype 3', type: 'TEXT' },
  { key: 'archetype_3_percent', name: 'Archetype 3 Percent', type: 'NUMBER' }
];
```



Enhanced Logging

The updated function now provides detailed console logs:

```
🔍 Checking custom fields...
📋 Existing fields: ['name', 'last_name', ...]
➕ Creating field: archetype
✅ Created field: archetype
➕ Creating field: archetype_percent
✅ Created field: archetype_percent
// ... more fields
✅ All custom fields are ready
📧 Adding to MailerLite: john@example.com
🎯 Archetype: The Sage (32%)
📊 Top 3: [{ name: 'The Sage', percentage: 32 }, ...]
📧 Sending to MailerLite: { ... }
📧 MailerLite response: { ... }
✅ Successfully added to MailerLite
```

These logs help you:

- See when fields are created (first run only)
- Verify data being sent
- Confirm successful API calls
- Debug any issues



Testing

Step 1: Deploy Updated Function

```
cd /home/ubuntu/archetypen_static
git add api/mailerlite.js
git commit -m "Fix MailerLite custom fields - auto-create if missing"
git push origin main
```

If using Vercel:

- Vercel will auto-deploy from GitHub
- Wait 2-3 minutes for deployment

Or deploy directly:

```
vercel --prod
```

Step 2: Complete a Test Assessment

1. Go to your deployed app
2. Fill in the intro form with a **test email**
3. Complete all 60 questions
4. Submit the assessment

Step 3: Verify in MailerLite

1. **Go to MailerLite Dashboard**
2. **Click “Subscribers”**
3. **Find your test email**
4. **Click on the subscriber**
5. **Scroll to “Custom Fields” section**

You should now see:

```
Archetype: The Sage  
Archetype Percent: 32  
Test Date: 1/14/2026  
Archetype 1: The Sage  
Archetype 1 Percent: 32  
Archetype 2: The Visionary  
Archetype 2 Percent: 24  
Archetype 3: The Builder  
Archetype 3 Percent: 18
```

Step 4: Verify Field Creation

1. **Go to MailerLite → Subscribers → Fields**
 2. **You should see all 9 custom fields:**
 - Archetype
 - Archetype Percent
 - Test Date
 - Archetype 1
 - Archetype 1 Percent
 - Archetype 2
 - Archetype 2 Percent
 - Archetype 3
 - Archetype 3 Percent
-

Troubleshooting

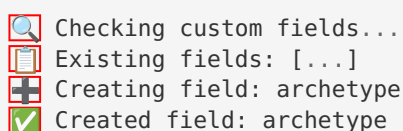
Issue: Fields still not showing

Check server logs:

If deployed on Vercel:

1. Go to Vercel Dashboard
2. Click on your project
3. Go to “Deployments” → Click latest deployment
4. Click “Functions” tab
5. Click on `api/mailerlite.js`
6. Check the logs

Look for:



```
Checking custom fields...
Existing fields: [...]
Creating field: archetype
Created field: archetype
```

Issue: “Failed to create field” error

Possible causes:

1. **API Key Invalid** - Check if `MAILERLITE_API_KEY` is correct
2. **API Permissions** - Ensure API key has permission to create fields
3. **Field Already Exists** - MailerLite may have created it with a different name



Solution:

- Manually check MailerLite Dashboard → Subscribers → Fields
- If fields exist with different names, update `REQUIRED_FIELDS` array in code

Issue: Subscriber added but fields empty

This means fields were created but data wasn’t sent properly.










Check:

1. Console logs:  `Sending to MailerLite:` - verify data structure
 2. Response logs:  `MailerLite response:` - check for errors
 3. Verify `top3Results` is being passed correctly from frontend
-

Behavior

First Run:






```

 Checking custom fields...
 Existing fields: ['name', 'last_name']
 Creating field: archetype
 Created field: archetype
 Creating field: archetype_percent
 Created field: archetype_percent
... (creates all 9 fields)
 All custom fields are ready
 Adding to MailerLite: john@example.com
 Successfully added to MailerLite

```

Subsequent Runs:






```

 Checking custom fields...
 Existing fields: ['name', 'archetype', 'archetype_percent', ...]
 All custom fields are ready (no creation needed)
 Adding to MailerLite: jane@example.com
 Successfully added to MailerLite

```

Note: Fields only need to be created once. After that, they're reused for all subscribers.

Benefits of Auto-Creation

1.  **No Manual Setup** - Fields are created automatically on first use
2.  **Always Up-to-Date** - If you add new fields, they're created automatically
3.  **Error-Proof** - No risk of forgetting to create a field
4.  **Portable** - Works on any MailerLite account without manual setup
5.  **Self-Healing** - If fields are deleted, they're recreated on next run

Using Custom Fields in Emails

Once fields are created and populated, you can use them in MailerLite emails:

```

<p>Hi {$name},</p>

<h2>Your Dominant Archetype: {$archetype}</h2>
<p>With {$archetype_percent}% alignment</p>

<h3>Your Complete Top 3:</h3>
<ol>
  <li>{$archetype_1} - {$archetype_1_percent}%</li>
  <li>{$archetype_2} - {$archetype_2_percent}%</li>
  <li>{$archetype_3} - {$archetype_3_percent}%</li>
</ol>

<p><small>Test completed on {$test_date}</small></p>

```





Summary

Problem: Custom fields weren't being populated in MailerLite

Cause: Fields didn't exist in MailerLite account

Solution: Updated serverless function to auto-create fields

Result:

-  Fields are created automatically
-  Data is populated correctly
-  No manual setup required
-  Works on any MailerLite account

Next Steps:

1. Deploy the updated code
2. Test with a real submission
3. Verify fields appear in MailerLite
4. Use fields in your email templates

The custom fields issue is now fixed! 