

Aside Analytics Implementation Guide

Pendo + Twilio Integration for Unified User Tracking

Executive Summary



What We're Implementing





This guide implements comprehensive analytics for Aside by configuring Pendo to track both SMS and web activity under unified user profiles. Users will be identified by phone number (E.164 format: +15555551234) across all platforms.

Current Problems

1. **Wrong User Identification**
 - 13 users identified by database IDs (31, 54, etc.) instead of phone numbers
 - 241 visitors (95%) showing as anonymous (*see "Inflated Visitor Counts" below for why this is misleading*)
 - Cannot link SMS activity to web activity
2. **Inflated Visitor Counts (Anonymous Duplicates)**
 - **Critical insight:** Those 241 anonymous visitors are NOT 241 unique people
 - Same person creates multiple anonymous IDs when they:
 - Clear cookies → New anonymous ID
 - Use different browsers (Chrome, Safari) → New anonymous ID per browser
 - Use different devices (laptop, phone) → New anonymous ID per device
 - Visit before logging in → New anonymous ID each visit
 - **Estimated reality:** 241 anonymous IDs = ~20-40 actual unique people
 - After implementing phone number visitor IDs, duplicates will consolidate into single profiles
3. **No SMS Tracking**
 - SMS messages (primary interaction) not tracked in Pendo
 - Missing significant user activity data - cannot measure SMS engagement or platform preference
4. **Issues Tagging Features in Pendo**
 - Cannot finish Pendo tagging until we've made the updates in this guide

After Implementation

-  All users identified by phone number (+15555551234)
-  SMS and web activity unified in single user profile

-  **Accurate visitor counts** - no more duplicate anonymous IDs
-  **Cross-device tracking** - same user ID on laptop, phone, any browser
-  **Persistent identification** - survives cookie clearing, works across sessions
-  **Clear insights into platform preference and feature adoption**

Expected Impact on Metrics

Current inflated metrics:

Total Visitors: 254 — Identified: 13 (5%) — Anonymous: 241 (95%)

After implementation (realistic metrics):

Total Visitors: ~30-50 actual unique people — Identified: 100% (everyone has phone number ID) — Anonymous: 0%

Duplicate IDs eliminated: — Same person across devices → ONE profile — Same person across browsers → ONE profile — Same person across sessions → ONE profile

Phase 1: Pendo Configuration Fixes






1. Fix Visitor Identification (CRITICAL)

Problem: Web users identified by database ID, not phone number. SMS tracking will use phone number, creating duplicate profiles.

Additional problem: Anonymous visitors are creating duplicate IDs for the same person across devices, browsers, and sessions, inflating visitor counts and making analytics meaningless.

Solution: Use phone number with +1 country code as visitor ID. This creates a permanent, universal identifier that works across all devices, browsers, and sessions.

Why phone number as ID solves the duplicate problem:

-  **Survives cookie clearing** - User identity not stored in cookies
-  **Works across devices** - Same phone number on laptop, phone, tablet
-  **Works across browsers** - Same ID in Chrome, Safari, Firefox
-  **Permanent identifier** - Never changes, unlike anonymous IDs
-  **Matches SMS tracking** - Twilio provides phone number, creates unified profile

Location: Find where `pendo.initialize()` is called (likely in authentication/login success handler)

Change from:

javascript

// WRONG - Database ID

```
pendo.initialize({
  visitor: {
    id: user.id, // 31, 54, 22, etc.
    phoneNumber: user.phone
  },
  account: {
    id: "context-app"
  }
});
```

Change to:

javascript

// CORRECT - Phone number as ID

```
pendo.initialize({
  visitor: {
    id: `+1${user.phone}`, // "+15555551234" - E.164 format
    phoneNumber: user.phone,
    firstName: user.firstName || undefined,
    lastName: user.lastName || undefined,
    userId: user.id, // Keep database ID as metadata
    hasCompletedProfile: !(user.firstName && user.lastName),
    signUpDate: user.createdAt,
    signUpMethod: user.signUpMethod || 'sms'
  },
  account: {
    id: "aside" // Updated from "context-app"
  }
});
```

Critical: Only call pendo.initialize() AFTER user authentication completes. Do not initialize before you know the user's phone number. If you initialize Pendo before login, it creates an anonymous visitor that won't merge with the identified user.

What happens to existing anonymous visitors:

- Old anonymous IDs will remain in Pendo (acceptable)
- They represent past sessions before this fix
- New activity will use phone number IDs
- Over time, anonymous visitors will age out of reports
- You'll see true visitor count emerge

Test:

1. Log out and clear browser
2. Log in with test account
3. Check Pendo → Visitors → Search for +1[your_phone]
4. Should show phone number as ID, NOT database ID or anonymous

Additional test - Cross-device:

1. Log in on desktop browser → Check Pendo visitor ID
 2. Log in on mobile browser → Check Pendo visitor ID
 3. Should be SAME visitor ID (phone number), not two separate visitors
-

2. Remove Click-Outside-to-Close from Modals (CRITICAL)

Problem: Modals close when clicking Pendo tagging tool, preventing product team from tagging modal features.

Solution: Remove backdrop click handlers entirely. Modals only close via Cancel button or X icon.

Location: Modal component code (React component or modal library config)

Remove this pattern:

```
javascript
// REMOVE - Backdrop click handler
const handleBackdropClick = (e) => {
  if (e.target === e.currentTarget) {
    onClose(); // ✗ Remove this
  }
};

return (
  <div className="modal-backdrop" onClick={handleBackdropClick}>
    <div className="modal-content">
      {children}
    </div>
  </div>
);
```

Replace with:

javascript

// CORRECT - No backdrop click handler

```
const Modal = ({ isOpen, onClose, children }) => {  
  // Optional: ESC key handler  
  useEffect(() => {  
    const handleEscapeKey = (e) => {  
      if (e.key === 'Escape') onClose();  
    };  
    if (isOpen) {  
      document.addEventListener('keydown', handleEscapeKey);  
      return () => document.removeEventListener('keydown', handleEscapeKey);  
    }  
  }, [isOpen, onClose]);  
  
  return (  
    <div className="modal-backdrop"> { /* No onClick */}  
      <div className="modal-content">  
        <button className="close-x" onClick={onClose}>×</button>  
        {children}  
        <button onClick={onClose}>Cancel</button>  
      </div>  
    </div>  
  );  
};
```

Apply to all modals.

Test:

1. Open any modal
2. Click outside modal → Should stay open
3. Click Pendo tagging tool → Modal stays open, can tag features

3. Update Board URL Structure

Problem: Want to better distinguish private vs shared boards by URL

Solution: Add board type to URL path.

Current URLs:

- /tag/recipes
- /tag/workouts

- /shared/home-improvement

New URLs:

- /tag/private/recipes
- /tag/private/workouts
- /tag/shared/home-improvement

Implementation:

Update routing:

javascript

// New route with board type

```
app.get('/tag/:boardType/:boardName', (req, res) => {
  const { boardType, boardName } = req.params;
  if (!['private', 'shared'].includes(boardType)) {
    return res.status(400).send('Invalid board type');
  }
  const board = getBoard(boardName);
  if (!board) {
    return res.status(404).send('Board not found');
  }
  // Verify URL type matches actual board type
  if (board.type !== boardType) {
    return res.redirect(301, `/tag/${board.type}/${boardName}`);
  }
  renderBoard(board);
});
```

// Redirect old URLs

```
app.get('/tag/:boardName', (req, res) => {
  const board = getBoard(req.params.boardName);
  if (!board) {
    return res.status(404).send('Board not found');
  }
  res.redirect(301, `/tag/${board.type}/${board.name}`);
});
```

Update all board links:

javascript

// BEFORE

```
<a href={`/tag/${board.name}`}>View Board</a>
```

// AFTER

```
<a href={` /tag/${board.type}/${board.name}`}>View Board</a>
```

Update locations:

- Dashboard board cards
- Sidebar board navigation
- Search results
- SMS response messages
- Any other board links

Test:

1. Navigate to /tag/recipes → Should redirect to /tag/private/recipes
 2. Create new private board → URL should be /tag/private/[name]
 3. Create new shared board → URL should be /tag/shared/[name]
-

4. Track Profile Updates in Real-Time

Problem: Pendo doesn't know when user updates profile until next login.

Solution: Call pendo.identify() immediately after profile save.

Location: Profile update/save handler

Add to profile save function:

javascript

```
async function handleProfileSave(formData) {  
  const { firstName, lastName, email } = formData;
```

// 1. Save to database (existing code)

```
  await updateUser(currentUser.id, {  
    firstName: firstName,  
    lastName: lastName,  
    email: email  
  });
```

// 2. ✨ NEW: Update Pendo immediately

```
  pendo.identify({  
    visitor: {  
      id: `+1${currentUser.phone}`, // Same phone number format  
      firstName: firstName,
```

```

    lastName: lastName,
    email: email,
    hasCompletedProfile: !(firstName && lastName),
    profileCompletedAt: new Date().toISOString()
  }
});

// 3. Show success message (existing code)
showSuccessToast("Profile updated!");
}

```

Test:

1. Log in and check Pendo visitor profile
2. Update first/last name in Edit Profile
3. Immediately check Pendo again (no logout/login needed)
4. Should show updated name

5. Remove Twilio Segment Integration

Problem: Segment adds complexity without value. Cannot track SMS, creates duplicate web tracking, events being filtered by Pendo.

Solution: Remove Segment completely.

Step 1: Remove Segment snippet from code

Search for and delete:

```

javascript
<script>
!function(){var analytics=window.analytics=window.analytics||[];
// ... long Segment snippet
analytics.load("YOUR_WRITE_KEY");
analytics.page();
}();
</script>

```

Step 2: Remove Segment tracking calls

Search codebase for and remove:

- analytics.track(

- analytics.page(
- analytics.identify(
- analytics.group(
- window.analytics

Keep Pendo calls:

- pendo.track(- KEEP
- pendo.initialize(- KEEP

Step 3: Remove Segment package

```
bash
npm uninstall @segment/analytics-next
# or whatever the package name is
```

Step 4: Remove environment variables

```
bash
# Remove from .env
SEGMENT_WRITE_KEY=xxxxx
SEGMENT_API_KEY=xxxxx
```

Step 5: Disable in Segment dashboard (Product team can do this)

1. Log into Segment
2. Connections → Destinations → "Aside Pendo" → Disable
3. Connections → Sources → "Context" → Disable

Test:

1. Open website in browser
2. DevTools Console → Type window.analytics → Should return undefined
3. Type window.pendo → Should return object
4. Network tab → No requests to cdn.segment.com

Phase 1.5: Implement Pendo Data Attributes

CRITICAL: Universal Requirement

Every button, link, input, modal, and interactive element in the application **MUST** have a unique **data-pendo** attribute. No exceptions.

If a user can click it, type into it, or interact with it in any way, it needs a **data-pendo** attribute. This is not optional - it's the foundation for all product analytics and feature tagging.

Why We Need This

Problem: Many interactive elements (buttons, links, forms) have identical HTML with no distinguishing attributes. This makes it impossible for the product team to reliably tag features in Pendo without using fragile selectors like **nth-child**.

Example - Identical buttons:

```
html
<!-- Private Boards + button -->
<button class="inline-flex items-center...">
  <svg>...</svg>
</button>

<!-- Shared Boards + button -->
<button class="inline-flex items-center...">
  <svg>...</svg>
</button>
```

Both buttons have the exact same HTML. Pendo can't tell them apart.

Solution: Add unique **data-pendo** attributes to all interactive elements. This creates stable, maintainable feature tags that won't break when CSS classes or HTML structure changes.

Modal Button Naming Strategy

Question: Multiple modals have "Cancel" buttons. Do they all need unique names like **convert-to-shared-cancel-btn**, **delete-board-cancel-btn**, etc.?

Answer: No! Use **generic button names** and let Pendo scope them by their parent modal container.

Recommended Approach: Generic Button Names

Use the **same data-pendo** value for all similar buttons across modals:

```
html
<!-- Convert to Shared Modal -->
```

```

<div data-pendo="convert-to-shared-modal">
  <h2>Convert to Shared Board</h2>
  <input data-pendo="invite-phone-input" placeholder="Phone number" />
  <button data-pendo="modal-cancel-btn">Cancel</button>
  <button data-pendo="modal-save-btn">Convert & Invite</button>
</div>

```

```

<!-- Delete Board Modal -->
<div data-pendo="delete-board-modal">
  <h2>Delete Board?</h2>
  <p>This action cannot be undone.</p>
  <button data-pendo="modal-cancel-btn">Cancel</button>
  <button data-pendo="modal-delete-confirm-btn">Delete</button>
</div>

```

```

<!-- Rename Board Modal -->
<div data-pendo="rename-board-modal">
  <h2>Rename Board</h2>
  <input data-pendo="board-name-input" />
  <button data-pendo="modal-cancel-btn">Cancel</button>
  <button data-pendo="modal-save-btn">Rename Board</button>
</div>

```

How Pendo Handles This:

When you tag a Cancel button in Pendo Visual Design Studio:

1. Click the Cancel button inside the Convert to Shared modal
2. Pendo automatically creates a **scoped selector** like:

CSS

```
[data-pendo="convert-to-shared-modal"] [data-pendo="modal-cancel-btn"]
```

3. This selector means: "The Cancel button **inside** the Convert to Shared modal"
4. It will **NOT** match Cancel buttons in other modals

Your Analytics Will Show:

Each button is tracked separately because Pendo knows the parent context:

- **Feature:** "Convert to Shared Modal - Cancel Button" → 23 clicks
- **Feature:** "Delete Board Modal - Cancel Button" → 12 clicks
- **Feature:** "Rename Board Modal - Cancel Button" → 45 clicks

Bonus: Aggregate Tracking

With generic button names, you can also create aggregate analytics:

- Create a feature for ALL `[data-pendo="modal-cancel-btn"]` clicks
- **Insight:** "Users clicked Cancel across all modals 80 times this week"
- Measures overall modal abandonment rate

Benefits of This Approach:

✅ **Simpler naming** - Reuse `modal-cancel-btn`, `modal-save-btn` everywhere ✅ **Still tracks separately** - Pendo scopes by parent modal container ✅ **Easier for developers** - Consistent, predictable pattern ✅ **Flexible analytics** - Can track individually OR in aggregate ✅ **Less code duplication** - Same button names across all modals

Alternative: Unique Names Per Modal

If you prefer completely unique button names:

html

```
<button data-pendo="convert-to-shared-cancel-btn">Cancel</button>
<button data-pendo="delete-board-cancel-btn">Cancel</button>
<button data-pendo="rename-board-cancel-btn">Cancel</button>
```

This works, but:

- ❌ More verbose naming
- ❌ More unique values to maintain
- ❌ Cannot track aggregate "all Cancel clicks"
- ✅ Simpler selectors in Pendo (no parent context needed)

We recommend generic names with parent modal scoping.

Naming Convention

Format: `data-pendo="[context]-[element-type]-[action]"`

Examples:

- `data-pendo="create-private-board-btn"`
- `data-pendo="create-shared-board-btn"`
- `data-pendo="delete-message-modal"`
- `data-pendo="board-rename-input"`

- `data-pendo="save-profile-btn"`

Rules:

- Use **lowercase with hyphens** (kebab-case)
 - Be **specific and descriptive**
 - Include **context** (which section/feature)
 - Include **element type** (btn, modal, input, link, etc.)
 - Include **action** when relevant (create, delete, edit, save)
-

What Needs Pendo Attributes

1. All Buttons

Every clickable button should have a `data-pendo` attribute:

html

```
<!-- Dashboard -->
<button data-pendo="dashboard-create-btn">+</button>
<button data-pendo="dashboard-search-btn">Search</button>

<!-- Board Management -->
<button data-pendo="create-private-board-btn">+</button>
<button data-pendo="create-shared-board-btn">+</button>
<button data-pendo="board-rename-btn">✎</button>
<button data-pendo="board-convert-to-shared-btn">👥</button>
<button data-pendo="board-delete-btn">🗑️</button>
<button data-pendo="board-members-btn">👁️</button>
<button data-pendo="board-invite-btn">👤+</button>

<!-- Content Actions -->
<button data-pendo="edit-message-btn">Edit</button>
<button data-pendo="delete-message-btn">Delete</button>
<button data-pendo="external-link-btn">→</button>

<!-- Modal Actions -->
<button data-pendo="modal-save-btn">Save</button>
<button data-pendo="modal-cancel-btn">Cancel</button>
<button data-pendo="modal-close-x-btn">×</button>
```

2. All Modals

Every modal container should be tagged:

html

```
<!-- Modal Containers -->
<div data-pendo="add-message-modal" class="modal">...</div>
<div data-pendo="create-private-board-modal" class="modal">...</div>
<div data-pendo="create-shared-board-modal" class="modal">...</div>
<div data-pendo="rename-board-modal" class="modal">...</div>
<div data-pendo="convert-to-shared-modal" class="modal">...</div>
<div data-pendo="delete-board-modal" class="modal">...</div>
<div data-pendo="board-members-modal" class="modal">...</div>
<div data-pendo="invite-to-board-modal" class="modal">...</div>
<div data-pendo="edit-message-modal" class="modal">...</div>
<div data-pendo="delete-message-modal" class="modal">...</div>
<div data-pendo="delete-account-modal" class="modal">...</div>
```

3. Navigation Elements

All navigation links and sections:

html

```
<!-- Sidebar Navigation -->
<nav data-pendo="sidebar-nav">
  <a href="/all-texts" data-pendo="nav-all-texts-link">All Texts</a>

  <!-- Private Boards Section -->
  <section data-pendo="private-boards-section">
    <h3>Private Boards</h3>
    <a
      href="/tag/private/drip"
      data-pendo="sidebar-board-link"
      data-board-name="drip"
      data-board-type="private"
    >
      #drip
    </a>
    <a
      href="/tag/private/recipes"
      data-pendo="sidebar-board-link"
      data-board-name="recipes"
      data-board-type="private"
    >
      #recipes
    </a>
```

```

<a
  href="/tag/private/feedback"
  data-pendo="sidebar-board-link"
  data-board-name="feedback"
  data-board-type="private"
>
  #feedback
</a>
</section>

<!-- Shared Boards Section -->
<section data-pendo="shared-boards-section">
  <h3>Shared Boards</h3>
  <a
    href="/tag/shared/home-improvement"
    data-pendo="sidebar-board-link"
    data-board-name="home-improvement"
    data-board-type="shared"
  >
    #home-improvement
  </a>
</section>
</nav>

<!-- Settings Navigation -->
<div data-pendo="settings-menu">
  <button data-pendo="settings-search-btn">🔍</button>
  <button data-pendo="settings-delete-account-btn">Delete Account</button>
  <button data-pendo="settings-notifications-btn">Notification Settings</button>
  <button data-pendo="settings-logout-btn">Logout</button>
</div>

```

Why sidebar board links use ONE data-pendo value:

Both private and shared board links in the sidebar use `data-pendo="sidebar-board-link"` (not separate values for private vs shared). This allows:

- ✅ Product team tags feature ONCE in Pendo using `[data-pendo="sidebar-board-link"]`
- ✅ Tracks ALL sidebar board navigation clicks (private and shared)
- ✅ Metadata (data-board-name, data-board-type) captured for segmentation
- ✅ Works for all current and future user-generated boards

Analytics Example:

- Feature: "Sidebar Board Link"
- Total clicks: 789
- Segment by `data-board-type`: 712 private, 77 shared
- Segment by `data-board-name`: 234 recipes, 189 drip, 123 feedback, etc.
- **Compare to:** Dashboard board card clicks to measure navigation preference (sidebar vs dashboard)

4. Form Elements

All form inputs and form containers:

html

```
<!-- Add Message Form -->
<form data-pendo="add-message-form">
  <textarea
    data-pendo="message-input"
    placeholder="Type your message..."
  ></textarea>
  <input
    data-pendo="message-tags-input"
    placeholder="#tags"
  />
  <button data-pendo="add-message-submit-btn">Add Message</button>
</form>

<!-- Board Creation Form -->
<form data-pendo="create-board-form">
  <input
    data-pendo="board-name-input"
    placeholder="Board name"
  />
  <button data-pendo="create-board-submit-btn">Create Board</button>
</form>

<!-- Profile Form -->
<form data-pendo="edit-profile-form">
  <input data-pendo="first-name-input" />
  <input data-pendo="last-name-input" />
  <input data-pendo="profile-picture-input" />
  <button data-pendo="save-profile-btn">Save Changes</button>
</form>
```

5. Content Cards

All content items and boards:

html

```
<!-- Board Cards on Dashboard -->
<!-- IMPORTANT: Tag with data-pendo ONLY, not board-specific selectors -->
<!-- This allows ONE Pendo feature tag to track ALL user-generated boards -->
<a
  href={` /tag/${board.type}/${board.name}`}
  data-pendo="dashboard-board-card"
  data-board-name={board.name}
  data-board-type={board.type}
  data-item-count={board.itemCount}
  className="board-card"
>
  <h3>#{board.name}</h3>
  <p>{board.type === 'private' ? 'Private' : 'Shared'} Board</p>
  <span>{board.itemCount}</span>
</a>

<!-- Content Item Cards -->
<div
  data-pendo="content-card"
  data-content-type="link"
  data-board-name="recipes"
  data-content-id="12345"
>
  <a href="https://example.com" data-pendo="content-title-link">
    Recipe Title
  </a>
  <button data-pendo="content-edit-btn">Edit</button>
  <button data-pendo="content-delete-btn">Delete</button>
</div>
```

Why this structure matters for user-generated content:

The dashboard board cards are **user-generated** - users create new boards dynamically. Using `data-pendo="dashboard-board-card"` for ALL board cards allows:

- ✅ Product team tags feature ONCE in Pendo using `[data-pendo="dashboard-board-card"]`
- ✅ Tracks clicks on ALL boards (existing and future user-generated ones)
- ✅ Metadata (data-board-name, data-board-type) captured automatically for segmentation
- ✅ No need to create new Pendo tags when users create new boards

Analytics Example:

- Feature: "Dashboard Board Card"
- Total clicks: 423
- Segment by **data-board-type**: 387 private, 36 shared
- Segment by **data-board-name**: 147 recipes, 89 drip, 67 feedback, etc.
- **Compare to**: Sidebar navigation clicks to measure navigation preference

6. Search Components

Search bar and results:

html


```
<!-- Search Bar -->
<div data-pendo="search-container">
  <input
    data-pendo="search-input"
    placeholder="Search..."
  />
  <button data-pendo="search-submit-btn"><img alt="search icon" data-bbox="465 435 485 455"/></button>
</div>

<!-- Search Results -->
<div data-pendo="search-results">
  <div
    data-pendo="search-result-item"
    data-result-position="1"
  >
    <a data-pendo="search-result-link">Result Title</a>
  </div>
</div>
```

Implementation Guidelines


For React Components:

jsx

```
//  CORRECT - Add data-pendo to component root
const CreatePrivateBoardButton = () => {
  return (
    <button
      data-pendo="create-private-board-btn"
      onClick={handleCreateBoard}
    />
  )
}
```

```


        className="inline-flex items-center..."
      >
        <PlusIcon />
      </button>
    );
  };

//  CORRECT - Dynamic data attributes for lists
const BoardCard = ({ board }) => {
  return (
    <div
      data-pendo="board-card"
      data-board-name={board.name}
      data-board-type={board.type}
      data-item-count={board.itemCount}
    >
      <h3>{board.name}</h3>
    </div>
  );
};

```

For Modal Components:

```

jsx
//  CORRECT - Tag modal container
const Modal = ({ isOpen, onClose, children, pendold }) => {
  if (!isOpen) return null;

  return (
    <div className="modal-backdrop">
      <div
        data-pendo={pendold} // Passed as prop
        className="modal-content"
      >
        <button
          data-pendo="modal-close-x-btn"
          onClick={onClose}
        >
          ×
        </button>
        {children}
      </div>
    </div>
  );
};

```

```
};
```

// Usage:

```
<Modal
  isOpen={showCreateBoard}
  onClose={closeModal}
  pendoId="create-private-board-modal"
>
  { /* Modal content */ }
</Modal>
```

For Dynamic Elements:

jsx

//  CORRECT - Combine base selector with dynamic data

```
<button
  data-pendo="nav-board-link"
  data-board-name={board.name}
  data-board-type={board.type}
>
  #{board.name}
</button>
```

// This allows Pendo to:

// 1. Tag ALL board links with [data-pendo="nav-board-link"]



// 2. Filter by specific board using data-board-name

// 3. Segment by board type using data-board-type

Testing Pendo Attributes

After implementing data attributes, test that Pendo can find them:

Test 1: Visual Design Studio

1. Open Pendo Visual Design Studio
2. Click on an element (e.g., the Create Private Board button)
3. Pendo should suggest: `button[data-pendo="create-private-board-btn"]`
4.  No `nth-child` warnings
5.  Clean, stable selector

Test 2: Browser Console

javascript

```
// Check that data attributes exist
document.querySelector('[data-pendo="create-private-board-btn"]')
// Should return: <button data-pendo="create-private-board-btn">...</button>

// Check all board cards have attributes
document.querySelectorAll('[data-pendo="board-card"]')
// Should return: NodeList of all board cards

// Check dynamic attributes exist
document.querySelector('[data-board-name="recipes"]')
// Should return the recipes board element
```

Test 3: Network Tab

1. Click a tagged button
 2. Check Network tab for Pendo tracking calls
 3. Should see event with feature name matching data-pendo value
-

Priority Implementation Order

Week 1: Critical Interactive Elements

- All modal containers (10 modals)
- All primary buttons (Create, Save, Cancel, Delete)
- Dashboard navigation elements

Week 2: Content & Board Elements

- Board cards on dashboard
- Content item cards
- Board management buttons (rename, convert, delete)
- Content action buttons (edit, delete, external link)

Week 3: Forms & Secondary Elements

- All form inputs and containers
 - Search components
 - Settings menu elements
 - Sidebar navigation links
-

Documentation for Product Team


After implementation, provide the product team with:

1. Data Attribute Reference Sheet Create a spreadsheet or doc listing:

- Element description
- data-pendo value
- Location in app
- Additional data attributes (if any)





2. Pendo Tagging Guide Show product team how to tag using the new attributes:

To tag the "Create Private Board" button in Pendo:





1. Open Visual Design Studio
 2. Click the + button next to "Private Boards"
 3. Pendo will suggest: button[data-pendo="create-private-board-btn"]
 4. Accept this selector 
-

Maintenance

When adding new features:




1.  Add **data-pendo** attribute to all new interactive elements
2.  Follow naming convention: [context]-[element-type]-[action]
3.  Update data attribute reference sheet
4.  Test in Pendo Visual Design Studio before deploying

When refactoring:

-  CSS classes can change freely
 -  HTML structure can change freely
 -  Never change or remove **data-pendo** attributes without coordinating with product team
 -  Changing a **data-pendo** value will break existing Pendo tags and guides
-

Success Criteria

After implementing Pendo data attributes:

-  Product team can tag any feature without developer help
-  No **nth-child** selectors in Pendo tags
-  Feature tags remain stable across code refactors

- ✓ All modals can be tagged without closing
 - ✓ Clear, descriptive selector names in Pendo Visual Design Studio
 - ✓ Every interactive element has a unique, stable identifier
-

Quick Reference for Developers

Always add **data-pendo** to:

- Buttons → **data-pendo="[context]-[action]-btn"**
- Modals → **data-pendo="[purpose]-modal"**
- Forms → **data-pendo="[purpose]-form"**
- Inputs → **data-pendo="[field-name]-input"**
- Links → **data-pendo="[context]-link"**
- Navigation → **data-pendo="nav-[destination]-link"**

Naming pattern: **[context]-[element-type]-[action]**

Examples:

- create-private-board-btn** ✓
 - btn_create_private_board** ✗ (use hyphens, not underscores)
 - createPrivateBoardBtn** ✗ (use lowercase, not camelCase)
 - button-1** ✗ (be descriptive, not generic)
-

Questions? Contact the product team before implementing if unclear on naming.

Phase 2: Twilio SMS Tracking

1. Install Pendo Server-Side SDK

In backend project directory:

```
bash
```

```
npm install pendo-track-events-node
```

2. Initialize Pendo in Backend

Location: Server initialization file (server.js, app.js, or index.js)

Add at top with imports:

```
javascript
const Pendo = require('pendo-track-events-node');

// Initialize with API key
const pendo = new Pendo({
  apiKey: process.env.PENDO_API_KEY
});

module.exports = { pendo };
```

Get Pendo API Key:

1. Log into Pendo
2. Settings → App Details → Copy "Integration Key"
3. Add to .env file:

```
bash
PENDO_API_KEY=your-integration-key-here
```

3. Add Tracking to Twilio Webhook

Location: /api/webhook/twilio endpoint

Current code structure:

```
javascript
app.post('/api/webhook/twilio', async (req, res) => {
  const { From, Body, MessageSid } = req.body;
  const urls = extractUrls(Body);
  const hashtags = extractHashtags(Body);
  const isNewBoard = await checkIfNewBoard(From, hashtags[0]);
  await saveMessage(From, Body, hashtags);
  if (isNewBoard && hashtags.length > 0) {
    await createBoard(From, hashtags[0]);
  }
  await sendConfirmationSMS(From, hashtags[0]);
  res.sendStatus(200);
});
```


Add Pendo tracking:

javascript

```
const { pendo } = require('./server');

app.post('/api/webhook/twilio', async (req, res) => {
  const { From, Body, MessageSid } = req.body;

  try {
    // Check if new user
    let user = await User.findOne({ phone: From });
    const isNewUser = !user;

    if (isNewUser) {
      user = await User.create({
        phone: From,
        signupMethod: 'sms',
        signupDate: new Date()
      });
    }

    // ✨ Identify new user in Pendo
    try {
      await pendo.identify({
        visitorId: From,
        visitor: {
          id: From,
          phone: From,
          signupDate: new Date().toISOString(),
          signupMethod: 'sms',
          phoneVerified: true
        }
      });
    } catch (error) {
      console.error('Pendo identify error:', error);
    }
  }

  // Parse message
  const urls = extractUrls(Body);
  const hashtags = extractHashtags(Body);
  const isNewBoard = hashtags.length > 0 ?
    await checkIfNewBoard(From, hashtags[0]) : false;

  // Save to database
```

```
await saveMessage(From, Body, hashtags);
```

```
// ⚡ Track SMS message in Pendo
```

```
try {  
  await pendo.track({  
    event: 'SMS Message Received',  
    visitorId: From, // Already in E.164 format from Twilio  
    properties: {  
      messageLength: Body.length,  
      containsUrl: urls.length > 0,  
      urlDomain: urls[0] ? new URL(urls[0]).hostname : null,  
      hashtagsUsed: hashtags,  
      hashtagCount: hashtags.length,  
      isNewHashtag: isNewBoard,  
      isNewUser: isNewUser,  
      timestamp: new Date().toISOString(),  
      source: 'sms'  
    }  
  });  
} catch (error) {  
  console.error('Pendo track error:', error);  
}
```

```
// Create board if new hashtag
```

```
if (isNewBoard && hashtags.length > 0) {  
  await createBoard(From, hashtags[0], 'private');
```

```
// ⚡ Track board creation
```

```
try {  
  await pendo.track({  
    event: 'New Board Created via SMS',  
    visitorId: From,  
    properties: {  
      boardName: hashtags[0],  
      boardType: 'private',  
      messageLength: Body.length,  
      containsUrl: urls.length > 0,  
      source: 'sms',  
      timestamp: new Date().toISOString()  
    }  
  });  
} catch (error) {  
  console.error('Pendo board track error:', error);  
}
```

```
}

// Send confirmation
await sendConfirmationSMS(From, hashtags[0]);

res.sendStatus(200);

} catch (error) {
  console.error('SMS webhook error:', error);
  res.sendStatus(500);
}
});
```

Key Points:

- Twilio sends From as +15555551234 (E.164 format) - use directly as visitorId
 - Matches web format: +1\${user.phone}
 - Wrap Pendo calls in try-catch - analytics should never break SMS processing
 - Start with 2 events: "SMS Message Received" and "New Board Created via SMS"
-

Testing Checklist

Phase 1 Tests

Test 1: Visitor Identification

- Log out and clear browser
- Log in with test account
- Check Pendo → Visitor ID is +1[phone], NOT database ID or anonymous

Test 1b: Cross-Device Consolidation

- Log in on desktop Chrome → Note Pendo visitor ID
- Log in on mobile Safari → Check Pendo visitor ID
- Should be SAME visitor ID (phone number), proving no duplicate IDs across devices

Test 2: Modal Tagging

- Open "Add to Aside" modal
- Click outside modal → Stays open
- Click Pendo tagging tool → Modal stays open
- Can tag features inside modal

Test 3: Board URLs

- Navigate to private board → URL is /tag/private/[name]
- Navigate to shared board → URL is /tag/shared/[name]
- Type old URL /tag/recipes → Redirects to new format

Test 4: Profile Updates


- Edit profile, change name
- Check Pendo immediately → Name updated (no logout needed)

Test 5: Segment Removal

- Browser console: window.analytics returns undefined
- Browser console: window.pendo returns object
- Network tab: No requests to segment.com

Test 6: Pendo Data Attributes

- Open browser console
- Run:

```
document.querySelector('[data-pendo="create-private-board-btn"]')
```
- Should return the button element
- Open Pendo Visual Design Studio
- Click the + button next to Private Boards
- Pendo should suggest: `button[data-pendo="create-private-board-btn"]`
-  No nth-child warnings

Phase 2 Tests

Test 7: New SMS User

- Text from new phone number
- Check Pendo → New visitor with phone number ID
- Shows event: "SMS Message Received"
- Shows event: "New Board Created via SMS"

Test 8: Existing SMS User

- Text from same phone again
- Check Pendo → SAME visitor (not new)
- New event added to timeline

Test 9: Unified Profile (CRITICAL)

- Text from phone: 5555551234

- Log into web with same phone
- Check Pendo for +15555551234
- Should see ONE profile with BOTH SMS and web events in timeline
- If two separate profiles exist, phone format doesn't match

Test 10: SMS with Pendo Down

- Set wrong Pendo API key temporarily
- Send SMS
- SMS still processes successfully
- Message saved, confirmation sent
- Pendo error logged but doesn't break SMS

Post-Implementation: Verify Accurate Visitor Counts

After Phase 1 is deployed and users start logging in:

Week 1 Check:

- Compare Pendo visitor count to your database user count (phone numbers)
- Should be much closer now (vs. 241 anonymous inflating numbers)
- New identified visitors should equal actual new signups

Week 2-4 Check:

- Monitor new anonymous visitor creation
- Should be near zero (only pre-login browsing if any)
- Identified visitor count should match database user count

What you'll see:

Before Fix:

- Pendo: 254 visitors (13 identified, 241 anonymous)
- Database: ~30-50 actual users
- Mismatch: Anonymous duplicates inflating count

After Fix (Week 4):

- Pendo: ~30-50 identified visitors
- Database: ~30-50 actual users
- Match: Accurate 1:1 correspondence

Environment Variables

Add to .env:

bash

PENDO_API_KEY=your-integration-key-from-pendo

Remove from .env (if present):

bash

SEGMENT_WRITE_KEY=xxxxx

SEGMENT_API_KEY=xxxxx

Deployment Sequence

Recommended order:

1. **Deploy Phase 1 changes together:**
 - Visitor ID fix
 - Modal fix
 - URL structure
 - Profile update tracking
 - Segment removal
 2. **Deploy Phase 1.5 (Pendo Data Attributes):**
 - Add data-pendo attributes to all interactive elements
 - Test with Pendo Visual Design Studio
 - Verify no nth-child selectors needed
 3. **Test Phase 1 + 1.5 in production** for 24-48 hours
 4. **Deploy Phase 2 (SMS tracking):**
 - Install Pendo SDK
 - Add webhook tracking
 - Monitor SMS processing
-

Success Criteria

After implementation:

- ✓ Pendo shows users with phone number IDs (not database IDs or anonymous)
- ✓ **Visitor count matches database user count** (no more inflated numbers from anonymous duplicates)
- ✓ **Cross-device tracking works** - same user ID on laptop, phone, any browser

- ✓ Product team can tag modal features independently
- ✓ Can distinguish private vs shared board usage by URL
- ✓ Profile updates reflect in Pendo immediately
- ✓ Segment completely removed
- ✓ **All interactive elements have data-pendo attributes**
- ✓ **Product team can tag features without developer help**
- ✓ **No nth-child selectors in any Pendo tags**
- ✓ **Dashboard board cards tracked holistically** (all user-generated boards)
- ✓ **Sidebar board links tracked holistically** (all user-generated boards)
- ✓ SMS messages tracked in Pendo
- ✓ Single user profile shows both SMS and web activity
- ✓ SMS processing works even if Pendo fails

Pendo Dashboard Example:

Visitor: +15555551234

Timeline: |— Oct 10, 2:00 PM - SMS Message Received (sms) |— Oct 10, 2:00 PM - New Board Created via SMS (sms) |— Oct 10, 3:00 PM - Dashboard Viewed (web) |— Oct 10, 3:01 PM - Dashboard Board Card Clicked (web) | |— Board: recipes, Type: private |— Oct 10, 3:02 PM - Board Viewed (web) |— Oct 10, 3:05 PM - Sidebar Board Link Clicked (web) |— Board: drip, Type: private

Visitor Count Validation:

Week 4 After Implementation:

- Pendo Identified Visitors: 47
- Database User Count: 47
- Anonymous Visitors: 0 ✓ Accurate 1:1 match

Feature Analytics Example:

Navigation Preference Analysis:

- Dashboard Board Card clicks: 423
 - Private boards: 387
 - Shared boards: 36
 - Top boards: recipes (147), drip (89), feedback (67)
- Sidebar Board Link clicks: 789
 - Private boards: 712
 - Shared boards: 77
 - Top boards: recipes (234), drip (189), feedback (123)

Insight: Users prefer sidebar navigation 65% of the time over dashboard cards

Quick Reference

Phone Number Format:

- Database stores: 5555551234
- Pendo uses: +15555551234 (add +1 prefix)
- Format: +1\${user.phone}
- **Why this matters:** Creates permanent ID that works across devices, browsers, sessions

Understanding Anonymous Visitor Duplicates:

- Same person clearing cookies = Multiple anonymous IDs
- Same person using laptop + phone = Multiple anonymous IDs
- Same person using Chrome + Safari = Multiple anonymous IDs
- **Solution:** Phone number ID consolidates all activity into one profile

Pendo Data Attributes:

- Format: `data-pendo="[context]-[element-type]-[action]"`
- Always use lowercase with hyphens (kebab-case)
- For user-generated content: Use ONE data-pendo value + metadata attributes
- Example: `data-pendo="dashboard-board-card" + data-board-name="recipes"`

Critical Files to Update:

- Login/authentication handler (pendo.initialize)
- Modal components (remove backdrop click)
- Routing configuration (add board type to URLs)
- Profile save handler (pendo.identify)
- Remove Segment snippet (HTML/JS)
- **ALL interactive elements (add data-pendo attributes)**
- **Dashboard board cards (add data-pendo + metadata)**
- **Sidebar navigation links (add data-pendo + metadata)**
- Twilio webhook (/api/webhook/twilio)

Support:

- Pendo docs: <https://docs.pendo.io>
- Twilio docs: <https://www.twilio.com/docs>