

File-Based Analytics Tracking Documentation

Overview

This document explains how to use, manage, and maintain the file-based analytics tracking system implemented for the MIRD website. The system captures user interactions and stores them in a JSON file on your server, allowing you to analyze user behavior without relying on third-party analytics tools.

How It Works

1. **Client-Side Collection:** The `analytics.js` file collects user interactions (page views, clicks, etc.)
2. **Server-Side Storage:** Data is sent to `track.php`, which appends it to a JSON file
3. **File Management:** The system includes automatic file rotation and error logging

File Structure

- `analytics.js` - Client-side tracking script
- `track.php` - Server-side endpoint that receives and stores data
- `analytics_data.json` - The file where tracking data is stored (created automatically)
- `analytics_errors.log` - Error log file (created if errors occur)

Security Features

The tracking system includes several security measures:

1. **Security Token:** Each request requires a token for validation
2. **CORS Protection:** Only specified domains can send tracking data
3. **IP Anonymization:** User IP addresses are anonymized for GDPR compliance
4. **Input Validation:** All incoming data is validated before storage

Managing Analytics Data

Viewing Data

The analytics data is stored in `analytics_data.json` in your website's root directory. Each line in this file is a complete JSON object representing one tracked event. You can:

1. Download the file via FTP/SFTP
2. View it directly on the server using tools like `less` or `cat`
3. Process it with tools like `jq` for filtering and analysis

Example of viewing recent analytics entries:

```
tail -n 20 analytics_data.json
```

Backing Up Data

It's recommended to back up your analytics data regularly:

1. **Manual Backup:** `bash cp analytics_data.json analytics_data_backup_$(date +%Y%m%d).json`
2. **Automated Backup** (add to crontab): `bash 0 0 * * 0 cp /path/to/analytics_data.json /path/to/backups/analytics_data_$(date +%Y%m%d).json`

Purging Data

To clear analytics data while maintaining the file:

```
echo "" > analytics_data.json
```

To completely remove and recreate:

```
rm analytics_data.json  
touch analytics_data.json  
chmod 664 analytics_data.json
```

Automatic File Rotation

The system automatically rotates the analytics file when it reaches 10MB:

1. The current file is renamed to `analytics_data.json.YYYY-MM-DD-HH-MM-SS.bak`
2. A new empty file is created to continue collecting data

You can adjust the maximum file size in the `track.php` file by modifying the `max_file_size` value.

Customization

Allowed Domains

To add or modify domains allowed to send tracking data, edit the `allowed_origins` array in `track.php`:

```
'allowed_origins' => [  
    'https://tglauner.com',  
    'https://www.tglauner.com',  
    // Add your domains here  
],
```

Security Token

To change the security token, update it in both files:

1. In `track.php`: `php 'security_token' => 'your_new_token_here'`
2. In `analytics.js`: `javascript 'X-Analytics-Token': 'your_new_token_here'`

Troubleshooting

If tracking data isn't being recorded:

1. Check the `analytics_errors.log` file for error messages
2. Verify that `track.php` has write permissions to the directory
3. Ensure the security token matches in both files
4. Check browser console for JavaScript errors
5. Verify that the user has given consent for analytics cookies

Data Analysis

The stored JSON data can be analyzed using various tools:

1. **Simple Counting:** `bash grep -c "pageview" analytics_data.json # Count pageviews`
`grep -c "CTA Click" analytics_data.json # Count CTA clicks`
2. **Using jq** (a command-line JSON processor): `bash cat analytics_data.json | jq 'select(.type=="pageview")' | wc -l # Count pageviews`
`cat analytics_data.json | jq 'select(.category=="Engagement")' | wc -l # Count engagement events`
3. **Export to CSV** (for spreadsheet analysis): `bash cat analytics_data.json | jq -r ' [.timestamp, .type, .category, .action, .label] | @csv' > analytics_export.csv`

Future Enhancements

Consider these potential enhancements:

1. Create a simple PHP dashboard to visualize the analytics data
2. Implement a data aggregation script to generate daily/weekly reports
3. Add more advanced tracking metrics based on your specific needs

Conclusion

This file-based analytics system provides a lightweight, privacy-focused way to track user interactions without relying on third-party services. By following the management guidelines in this document, you can maintain an effective analytics system while keeping full control of your data.