

Bug Report

Bug ID: BUG-API-001

- **Title:** API Call Fails Due to Deprecated `mobileview` Action
- **Severity:** Critical
- **Priority:** High
- **Environment:**
 - App Version: [14]
 - Device: [Real Me 9i 5g]
- **Description:**

When attempting to load content on the app, the API call to Wikipedia's `w/api.php` fails due to an unrecognized value for the `action` parameter (`mobileview`). This results in the page displaying an error message, breaking the app's functionality.
- **Steps to Reproduce:**
 - Open the Wikipedia app.
 - Navigate to the "Main Page."
 - Click on any content.
- **Expected Result:**

The app should load the content seamlessly using the API.

Actual Result:

The app fails to load content. The API response is as follows:

json

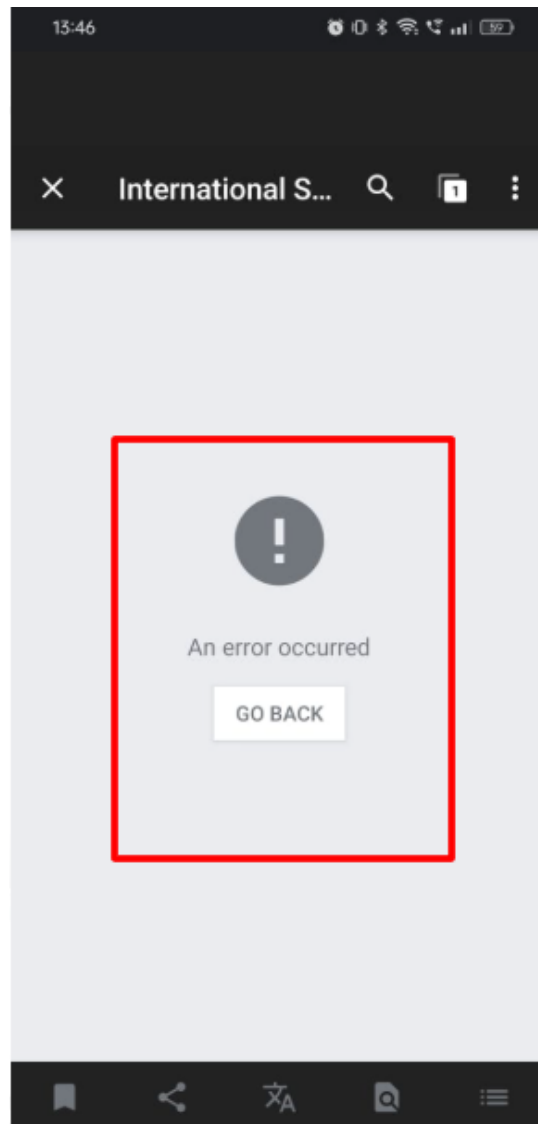
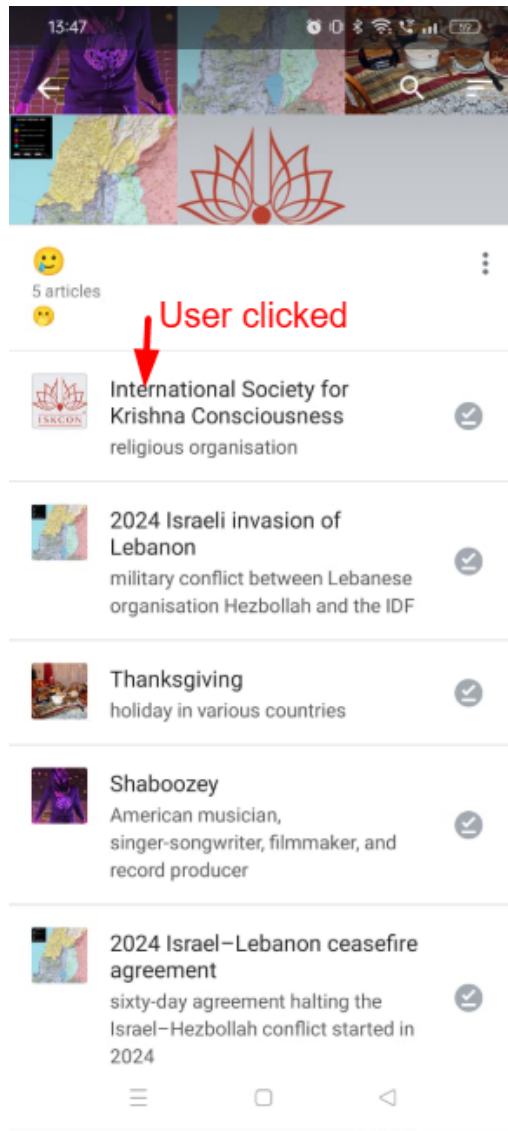
Copy code

```
{
  "error": {
    "code": "badvalue",
    "info": "Unrecognized value for parameter \"action\":
mobileview.",
    "docref": "See https://en.wikipedia.org/w/api.php for API
usage."
  },
  "servedby": "mw-api-ext.codfw.main-74f675c4bb-8ck6b"
}
```

-
- **Possible Cause:**

The `mobileview` action appears to be deprecated or removed from the Wikipedia API. The app may need to use an updated `action` parameter, such as `parse` or another supported action, as per the latest API documentation.
- **Suggested Fix:**

- Update the API call in the app to use a valid **action** parameter based on the latest Wikipedia API documentation.
- Test the updated API request to ensure compatibility and functionality.



Bug ID: BUG-REG-001

- **Title:** User Registration Fails Due to Missing "createtoken" Parameter and Poor Validation Feedback
- **Severity:** High
- **Priority:** Critical

Description:

During the registration process, the API (`createaccount`) fails due to a missing "createtoken" parameter. Additionally, the registration page lacks real-time client-side validations for username availability, password strength, and CAPTCHA correctness. Users only receive backend error messages, leading to confusion and a poor registration experience.

Steps to Reproduce:

1. Open the registration page.
2. Enter any username (e.g., "Mohi8").
3. Enter a password less than 8 characters.
4. Solve the CAPTCHA and submit the form.

Expected Results:

1. The form should validate username availability and password strength in real-time.
2. If CAPTCHA is incorrect, the form should provide a user-friendly error message.
3. The API should execute successfully with all required parameters, including "createtoken".

Actual Results:

1. No real-time validation for username or password.
2. Backend API responds with errors:

```
{  
  
  "error": {  
    "code": "missingparam",  
    "info": "The \"createtoken\" parameter must be set."  
  }  
}
```

3. API response also provides detailed failure messages for username and password, but these are not displayed on the UI.

Possible Causes:

1. The "createtoken" parameter is missing from the API request.
2. No client-side validation logic for form inputs.
3. Backend error messages are not processed or displayed in the UI.

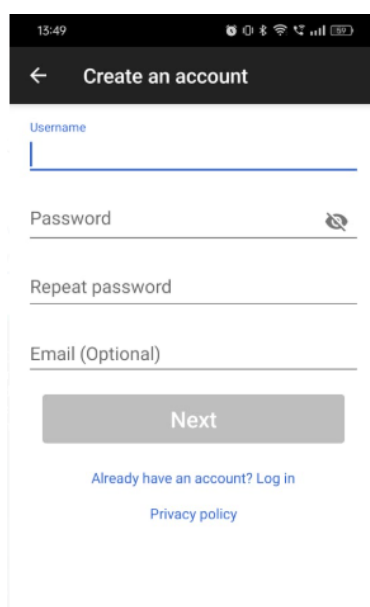
Impact:

This bug blocks:

1. Testing of the login functionality (dependent on user accounts).
2. Any workflows requiring user accounts, such as profile setup or personalized features.

Suggestions for Fix:

1. **API Implementation:**
 - Include the "createtoken" parameter in the API request payload.
 - Ensure the API properly handles all parameters.
2. **UI/UX Enhancements:**
 - Implement client-side validations for:
 - Username availability (real-time check with the API).
 - Password strength requirements.
 - CAPTCHA correctness.
 - Display meaningful error messages for each validation failure.
3. **Backend Improvements:**
 - Return more user-friendly error messages if the "createtoken" is missing.



13:49

← Create an account

Username

Password

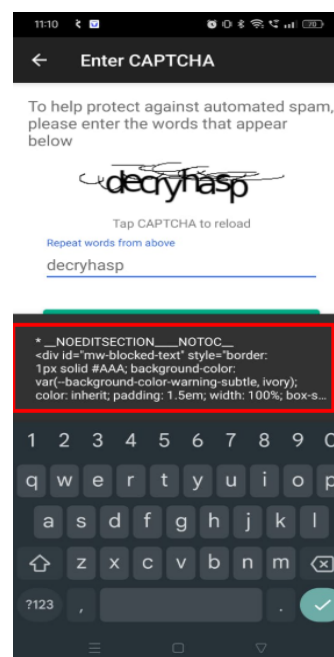
Repeat password

Email (Optional)

Next

Already have an account? Log in

Privacy policy



11:10

← Enter CAPTCHA

To help protect against automated spam, please enter the words that appear below

decryhasp

Tap CAPTCHA to reload

Repeat words from above

decryhasp

* __NOEDITSECTION__NOTOC__
<div id="mw-blocked-text" style="border: 1px solid #AAA; background-color: var(--background-color-warning-subtle, ivory); color: inherit; padding: 1.5em; width: 100%; box-s...

1 2 3 4 5 6 7 8 9 0

q w e r t y u i o p

a s d f g h j k l

↑ z x c v b n m ↵

?123 , . ✓

Bug ID: BUG-NEARBY-001

- **Title:** Invalid `ggsradius` Parameter Passed to API Causes "Current Location" Button to Fail
- **Severity:** High
- **Priority:** Critical

Description:

The "Current Location" button on the "Nearby" page fails because the application sends a floating-point value (`5458.177249585137`) for the `ggsradius` parameter, whereas the API expects an integer. This results in an API error (`badinteger`), and the UI displays an error message.

Steps to Reproduce:

1. Navigate to the "Nearby" page in the app.
2. Click the "Current Location" button.
3. Observe the error message on the UI.
4. Inspect the API request and response for details.

Expected Results:

- The app should pass a valid integer value for the `ggsradius` parameter.
- The API request should succeed, and the app should display nearby results.

Actual Results:

- The app sends a floating-point value (`5458.177249585137`) for the `ggsradius` parameter.

The API responds with an error:

json

Copy code

```
{
  "error": {
    "code": "badinteger",
    "info": "Invalid value \"5458.177249585137\" for integer parameter \"ggsradius\"."
  }
}
```

-

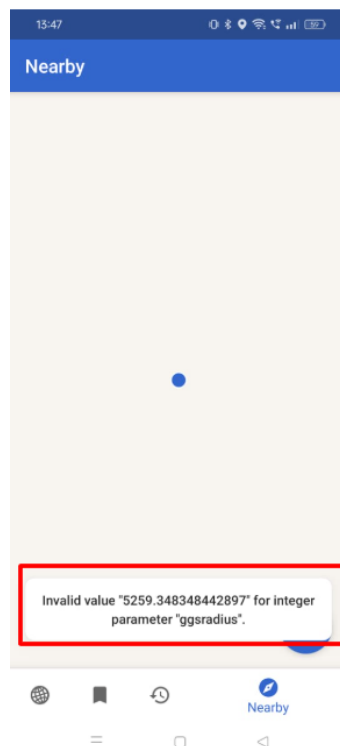
- The UI displays: "Invalid value 5458.177249585137 for integer parameter ggsradius."

Possible Causes:

1. The app calculates or retrieves a location radius value as a floating-point number but does not format it to an integer before sending the API request.

Suggestions for Fix:

1. **Frontend Validation:**
 - Ensure the `ggsradius` value is always converted to an integer before sending the API request.
 - Implement a rounding or truncation function to handle such scenarios.
2. **Error Handling on UI:**
 - Instead of displaying the raw API error to the user, provide a user-friendly error message such as:
"Unable to fetch nearby locations. Please try again later."
3. **API Error Handling:**
 - The backend could provide a clearer message or accept floating-point values by internally rounding them to integers



Bug ID: BUG-LOGIN-001

- **Title:** Missing Google Login Option in the Wikipedia Clone App
- **Severity:** Medium
- **Priority:** High

Description:

The login page of the Wikipedia clone app does not provide an option for users to log in via Google. This limits user convenience and excludes a widely preferred login method.

Steps to Reproduce:

1. Open the Wikipedia clone app.
2. Navigate to the "Login" page.
3. Look for a Google login option on the page.

Expected Results:

The app should provide an option for users to log in using their Google accounts for enhanced accessibility and user experience.

Actual Results:

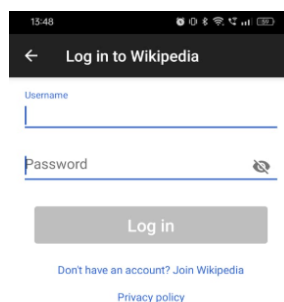
There is no Google login option available on the login page.

Possible Causes:

1. Google login integration has not been implemented.
2. The login options were not updated during development.

Suggestions for Fix:

1. **Frontend Update:** Add a "Google Login" button to the login page.
2. **Backend Implementation:** Integrate Google OAuth 2.0 for authentication.
3. **Error Handling:** Ensure proper error handling if the Google login fails



Bug ID: BUG-REG-002

- **Title:** Error Message on Registration Page Displays Raw HTML Code
- **Severity:** High
- **Priority:** Medium

Description:

When a user submits invalid details during registration (e.g., leaves the username blank or enters an invalid password), the error message displayed in the UI contains raw HTML tags (`<div>` and ``), making it difficult for users to understand the issue.

Steps to Reproduce:

1. Open the Wikipedia clone app.
2. Navigate to the "Registration" page.
3. Leave the "Username" field blank and provide an invalid password.
4. Solve the CAPTCHA correctly and click "Submit."
5. Observe the error message displayed.

Expected Results:

The error message should be properly formatted and displayed without any raw HTML tags.

Actual Results:

The error message contains raw HTML tags, making it appear unformatted and confusing for the user.

Possible Causes:

1. The backend error message is being sent as raw HTML without proper sanitization or parsing.
2. The frontend is not interpreting or rendering the error message correctly.

Suggestions for Fix:

1. **Backend Fix:** Ensure error messages are sent as plain text or JSON, not as raw HTML.
2. **Frontend Fix:** Parse and display error messages in a user-friendly format.
3. **Testing:** Validate error message formatting during the QA phase for all scenarios.

11:10

← Enter CAPTCHA

To help protect against automated spam,
please enter the words that appear
below

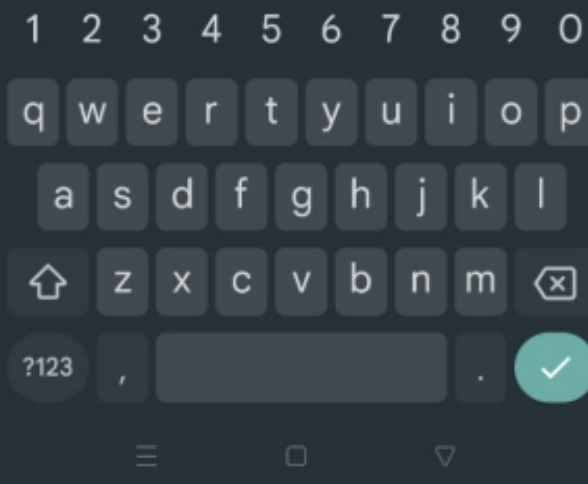


Tap CAPTCHA to reload

[Repeat words from above](#)

decryhasp

```
* __NOEDITSECTION__ __NOTOC__  
<div id="mw-blocked-text" style="border:  
1px solid #AAA; background-color:  
var(--background-color-warning-subtle, ivory);  
color: inherit; padding: 1.5em; width: 100%; box-s...
```



Bug ID: BUG-UI-001

- **Title:** Missing "Scroll to Top" Button in the Wikipedia Clone App
- **Severity:** Low
- **Priority:** Medium

Description:

The app does not provide a "Scroll to Top" button for users to quickly navigate back to the top of the page. This is a common and expected feature in content-heavy applications and its absence affects the user experience, especially on pages with long content.

Steps to Reproduce:

1. Open the Wikipedia clone app.
2. Navigate to a page with large content requiring scrolling.
3. Scroll to the bottom of the page.
4. Observe the absence of a "Scroll to Top" button or similar functionality.

Expected Results:

A "Scroll to Top" button should be available and visible at the bottom of the page. Clicking on it should smoothly navigate the user to the top of the page.

Actual Results:

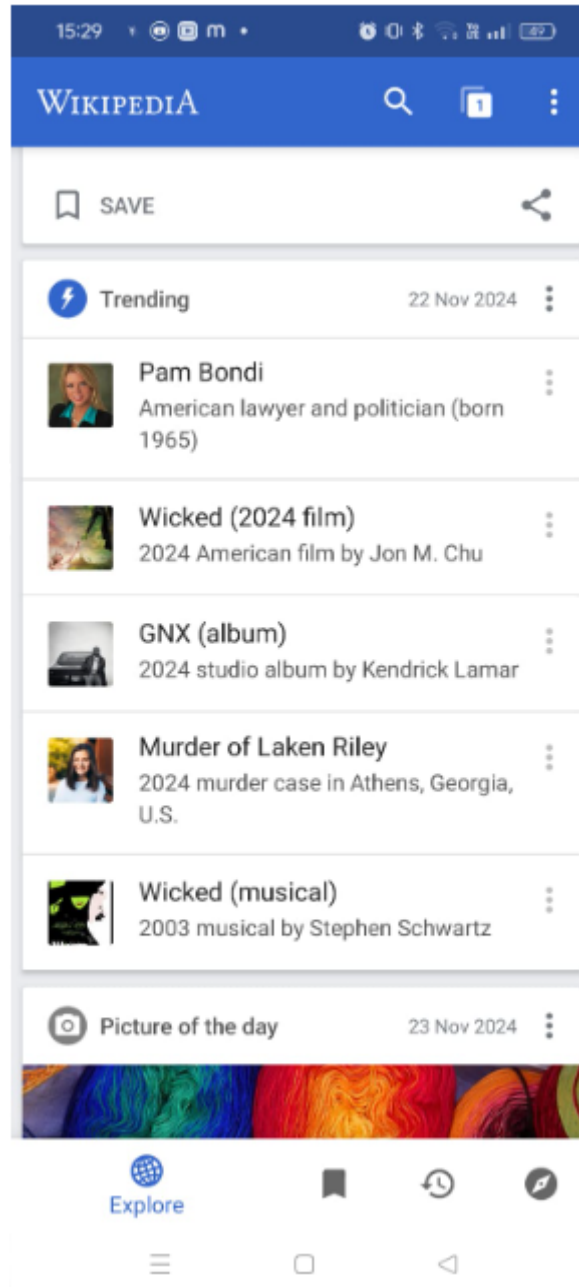
The app does not display any "Scroll to Top" button, forcing users to scroll manually to the top.

Possible Causes:

1. The feature has not been implemented during development.
2. The app design might have overlooked this functionality for user convenience.

Suggestions for Fix:

1. **Frontend Implementation:** Add a "Scroll to Top" button that appears when the user scrolls down a significant distance.
2. **UX Design:** Ensure the button is non-intrusive, typically placed in the bottom-right corner of the screen.
3. **Testing:** Validate the functionality across different screen sizes and devices.



Bug ID: BUG-REG-003

- **Title:** Email Field in User Registration is Optional Instead of Mandatory
- **Severity:** Medium
- **Priority:** High

Description:

The email field in the user registration process is currently optional, which is not aligned with standard registration requirements. In many applications, an email is a necessary field for account validation, communication, and password recovery. This issue could lead to incomplete user registration.

Steps to Reproduce:

1. Open the Wikipedia clone app.
2. Navigate to the "Registration" page.
3. Fill in the required fields but leave the "Email" field blank.
4. Solve the CAPTCHA correctly and click "Submit."
5. Observe that the registration process continues without any error message for the email field.

Expected Results:

1. The email field should be mandatory.
2. An error message should appear if the email is left blank: "Email is required."
3. The registration process should not allow submission without the email field being filled.

Actual Results:

1. The email field is optional.
2. No error message appears when the field is left blank.

Possible Causes:

1. The email field is incorrectly marked as optional in the backend.
2. The validation logic on the frontend is not correctly implemented for this field.

Suggestions for Fix:

1. **Frontend Fix:** Make the email field mandatory and add an appropriate error message when left blank.
 2. **Backend Fix:** Ensure that the registration API checks if the email field is provided and returns an error if it is missing.
 3. **Testing:** Validate the error message and registration flow after fixing the issue.
-

Bug ID: BUG-REG-004

- **Title:** Missing UI Validation Messages for Registration Form
- **Severity:** High
- **Priority:** Medium

Description:

The registration form does not display validation messages for mandatory fields such as username and password. Although the backend correctly returns validation messages (e.g., for a password that is too short or a username that is too similar), these messages are not visible to the user in the UI. This leads to confusion as the user is unaware of why the form is not submitting.

Steps to Reproduce:

1. Open the Wikipedia clone app.
2. Navigate to the "Registration" page.
3. Enter an invalid username (e.g., "Mohi8" which is too similar to another username "Moh18").
4. Enter a password that does not meet the criteria (e.g., less than 8 characters).
5. Solve the CAPTCHA correctly.
6. Click the "Submit" button.
7. Observe that there are no validation messages displayed in the UI, even though the backend has returned validation errors.

Expected Results:

1. The UI should display error messages, matching the backend's validation message.
2. Clear instructions should be provided, such as:
 - "The username 'Mohi8' is too similar to 'Moh18'. Please choose another username."
 - "Passwords must be at least 8 characters."
3. The form should not submit until all validation criteria are met.

Actual Results:

1. No validation messages are displayed on the UI, leading to user confusion.
2. The user is unable to know the exact reason why the form isn't being submitted.

Backend Response:

json

Copy code

```
{  
  "createaccount": {  
    "status": "FAIL",
```

```
"message": "* Passwords must be at least 8 characters.\n* The\nusername \"Mohi8\" is too similar to the following username:\n\n* Moh18\n\nPlease choose another username."
}
```

Possible Causes:

1. The frontend is not properly receiving or displaying the validation error messages from the backend.
2. Missing UI components for displaying form validation errors (e.g., labels or alert popups).

Suggestions for Fix:

1. **Frontend Fix:** Implement proper UI validation components that will show the validation error messages returned from the backend.
 2. **Backend Fix:** Ensure that the backend sends detailed validation error messages for both username and password.
 3. **User Experience Improvement:** Display the validation messages near the relevant form fields (username, password) to make it clear to the user what needs to be corrected.
-