CHECKOUT

Version: 17 March 2024

Reputation scoring on CheckOut

In our ongoing commitment to enrich user experience and uphold rigorous quality standards, CheckOut has implemented a sophisticated algorithm to meticulously assess submissions. This policy serves to elucidate the intricacies of our data collection process and its consequential impact on the rankings and content moderation mechanisms within our platform. Continuously evolving, we dedicate ourselves to refining this algorithm to better cater to the diverse needs of our user community, thereby fostering an environment of transparency, integrity, and inclusivity across CheckOut.

Definitions:

- rejectScore: The rejectScore is the ultimate reputation score assigned to a code. It's not generated upon code submission but instead calculated in real-time when codes are requested for display on CheckOut's website or apps, or within the API.

- Confidence score: The confidence score is a visual rating assigned to codes based on their rejectScore. These ratings may be showcased in webviews and the CheckOut app. Additionally, if accessible, they will be provided in the codes and active API.

- Submission: refers to the utilisation of the submit POST API, which is employed beneath the submission forms showcased across CheckOut's website and apps. Engaging with the API manually outside of the official submission forms is strongly discouraged and may lead to penalties, as elaborated below.

- Flagged: denotes a status of code within the database that is deemed non-valid. This status can arise due to various reasons, such as community reports of incorrectness, manual moderation, submission attempts exceeding rate limits, or automatic flagging by our system. Automated determinations categorising code as non-valid negatively impacts your reputation, and users will encounter an error message upon attempting to submit such code.

- Association: CheckOut acknowledges that IP addresses might be shared among multiple users. However, based on our analysis of existing logs, we've observed IP sharing to be so rare that it remains logical to assess scores based on past IP submissions that are associated.

Algorithm:

- Comparison with Past Submissions: If either you or another user has previously submitted a successful code from the same IP address, your score increases by 1 point. However, if the previous submission from that IP address was flagged, the code score decreases by 5 points. This process operates on a linear scale. For instance, if there were 12 successful submissions in the past and 1 flagged submission, the resulting score would be 7 (12 - 5).

- Accounts: Applying the same rationale to past submissions from the same IP, submissions made from the same user account earn 1 point for each successful submission and lose 5 points for flagged submissions. This scoring system works in conjunction with the IP logic. Therefore, having both a user account and a consistent IP address yields double the reputation impact compared to having only one tracking method.

- Moderation Accounts: If your account is designated as a moderation account, you receive a flat increase of 10 points per past submission. This bonus is added on top of the IP and all user scoring logic. In practical terms, if a moderation account has made 15 successful past submissions, they gain 15 points, and an additional 150 points (15 x 10) due to having a moderation account.

- Reputation Impact: Crowdsourced reporting also influences the score. For each reputation point you've received, your score decreases by that amount. For instance, if 5 users reported one of your associated past submissions, and another 3 users reported another one of your associated past submissions, your score would decrease by 7 (- (3 + 5)). Reporting functionality varies across classes and may not always be available to end users. This feature is new, and we are evaluating the effectiveness of user-reported incorrect codes.

Penalties:

To safeguard CheckOut spam-reduction methodologies, this section is brief, however it aims to provide an outline of how to avoid automated penalties.

- Invalid codes: Codes with leading zero(s) or likely spam. Score decreases by 5.
- Banned terms and characters in the groupCode, and other elements of the submission. Score decreases by 5.
- Duplicate submissions, or submission re-attempts. Score decreases by 5. Be careful when using the webform, if you encounter an error - read it and do not keep trying to submit the code, as every submission attempt is a dedication of 5 points.
- Attempts to use the submission AI with invalid tokens, or outside of the web and app forms. Score may be decreased by up to 10 per attempt, and automated and/or manual banning/rate limiting of the user may ensue.

Personal data use and security:

Codes submitted to CheckOut are kept anonymous to other users. However, we understand the concerns about moderators having access to users' IPs, user-agents, and other technical information. It's worth noting that logging technical information is a common practice for internet applications, including CheckOut. While we could implement a hashing algorithm to reduce the exposure of users' IPs to moderators, it's important to recognize that this wouldn't prevent technical administrators from accessing log data containing detailed technical information.

Access to this technical data is restricted by secure authentication measures. Moderators, who are trusted members of the community, have personal logins to access moderation tools. They play a vital role in maintaining the integrity of the platform.

We value your thoughts on data protection. If you have any suggestions or feedback, we're eager to hear from you.