HYPERTEST

SOLUTION BRIEF

THE KIND OF APPLICATION ERRORS THAT WILL HAPPEN BECAUSE OF API FAILURES

HYPERTEST

WHAT KIND OF API ERRORS CAN HYPERTEST CATCH ?

List of high severity errors HyperTest will never miss

CONTENT TYPE CHANGES

Content type header has changed in the new version

KEY REMOVED

Contract failure with removal of Key in the response object

DATA TYPE CHANGE

Contract schema remains similar, but data type changes i.e. integer to string etc

ARRAY ORDER CHANGE

Order of data received is different

VALUE MODIFIED

Contract schema remains similar, but data changes

SLOWENESS IN APIs

Change in Latency or Bandwidth wrt stable app

SENSITIVE DATA EXPOSURE

Leakage of critical user details in APIs that shouldn't

STATUS CODE CHANGE

Fatal crashes with status code failures 400s, 500s etc

SCAN TO SCHEDULE A DEMO WITH OUR TEAM

ERROR TYPE: CONTENT TYPE CHANGE	
SEVERITY	**Very High**
WHAT DOES THIS ERROR MEAN?	Content type header has changed in the new version
EXAMPLE	The old app is returning a JSON and new app is returning HTML
WHY DOES THIS USUALLY HAPPEN?	This is rarely desired, content types arent supposed to be updated every now and then. This error mostly happens when a service further downstream is down/buggy and response is short circuited by a proxy or another service.
POSSIBLE NEGATIVE IMPACTS	Upstream/Client App would crash. It wont be able to understand the new response and throw an exception
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. Check if the Client/upstream has been updated to handle new content type 2. Have a strategy for forcing an update of old clients/upstreams before deploying new release 3. If force update is not an option, implement API versioning for gradual update of clients and deprecate once every client is updated or no longer supported

ERROR TYPE: VALUE / DATA TYPE MODIFIED	
SEVERITY	**Very High**
WHAT DOES THIS ERROR MEAN?	Contract schema remains similar, but data or data type changes
EXAMPLE	The old version calculates discount 10% discount on an item as $15, new version calculates the disocunt as $17.673535836 for the exact same case
WHY DOES THIS USUALLY HAPPEN?	1. Current timestamp dependent data (autoignored by HyperTest in omst cases, can be ignored safely) 2. Bug in new implementation 3. Data type changes in schema
POSSIBLE NEGATIVE IMPACTS	1. Incorrect data would be reported to upstream/client app due to flawed business logic 2. Upstream/client app can also crash if modified data is out of bounds
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. Verify new business logic 2. Check if upstream/client app is handling new data types properly. Eg: If float point numbers are sent to a client which earlier catered to integers, client would crash or drop some data leading to data issues

ERROR TYPE: STATUS CODE CHANGE	
SEVERITY	**Very High**
WHAT DOES THIS ERROR MEAN?	Fatal crash in the app reflected by a status code change
EXAMPLE	Old app is returning 200, and new app its returning 4xx or 5xx
WHY DOES THIS USUALLY HAPPEN?	1. Infra problems: Service/dowstream is down 2. Authentication logic is broken (for 401 errors) 3. Rate limits is exceeded by HyperTest (for 429) 4. Theres a bug in the service 5. Non idempotent action is being performed without making necessary configurations in HyperTest first
POSSIBLE NEGATIVE IMPACTS	Upsteam/Client App would crash. If exceptions are handled, upstream/client app may throw a generic exception to the user. They wont be able to perform the action that they meant to do
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. If this happens even without a code change, it must be a non idempotent request. Handle this the "non-idempotency" by using HyperTest middlewares 2. If ratelimit is causing problem, add HyperTest IP to the whitelist for ratelimits 3. If infra related issue, debug and get that fixed

ERROR TYPE: KEY REMOVED	
SEVERITY	High
WHAT DOES THIS ERROR MEAN?	Fatal contract failure in the app due to removal of a key in the response
EXAMPLE	Rating Key removed
WHY DOES THIS USUALLY HAPPEN?	1. Change in logic to reduce the information that is shown to the user 2. Remove unused keys on upstream/client app to conserve bandwith and reduce response time/parse time on client 3. Bug in service 4. Unhandled dowstream contract changes in upstream
POSSIBLE NEGATIVE IMPACTS	1. Upstream/client app would crash if required keys are removed from the response 2. Upstream might fall back to default logic in absense of required keys which might result in wrong data problems. For example: If "discountPercentage" is removed from discount service response, cart service might crash or start giving a default discount of 0% which would not let the user avail the correct discount amount
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. Verify why was the key removed 2. Check if upstream/client app has been updated for removal of this key. 3. Check if there is no defaults for handling this key that might result in a change in business logic resulting in bugs

ERROR TYPE: KEY ADDED	
SEVERITY	Medium
WHAT DOES THIS ERROR MEAN?	Fatal or non-fatal contract failure in the app due to addition of a new Key
EXAMPLE	Old version does not reveal phone no of the user in the new version phone no is revealed
WHY DOES THIS USUALLY HAPPEN?	1. New feature in upstream/client app 2. Extra data coming from downstream is passed on without restriction to upstream 3. Fetching all data in a single query
POSSIBLE NEGATIVE IMPACTS	1. Unnecessary extra data might reveal sensitive user information 2. Extra data increases costs, bandwidth and response times
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. Follow principal of least priviliges, only reveal the minium data that is required by your upstream/client app 2. Report this as issue for non necessary keys

ERROR TYPE: ARRAY ORDER CHANGE	
SEVERITY	Medium
WHAT DOES THIS ERROR MEAN?	Order of data received is different
EXAMPLE	Users listed by ratings are shown in the wrong order in the new version
WHY DOES THIS USUALLY HAPPEN?	1. No order enforced on downstreams 2. Change in sorting logic 3. Bug in queries that fetch data
POSSIBLE NEGATIVE IMPACTS	1. Users would be reported incorrect data or in the wrong priority This might be a huge problem for stricty ordered data, timeseries data, etc 2. Orderless data can lead to cache misses and lead to more costs, bandwidth and response times
SUGGESTED ACTIONS FOR WARRANTED CHANGE	1. If the reported data should be sorted in a particular order, this is a huge bug 2. Ignore otherwise

ERROR TYPE: HEADER ADDED

SEVERITY	Low
WHAT DOES THIS ERROR MEAN?	Fatal or non-fatal contract failure in the app due to addition of a new header
EXAMPLE	Access Control methods header added
WHY DOES THIS USUALLY HAPPEN?	1. Misconfigured proxy settings 2. Misconfigured security settings in application
POSSIBLE NEGATIVE IMPACTS	1. Extra information regarding app/server might be revealed to a patential attacker 2. Extra bandwidth costs for both client and server
SUGGESTED ACTIONS FOR WARRANTED CHANGE	Verify why the reported headers are added. If they don't fulfil a purpose, get them removed

ERROR TYPE: HEADER MODIFIED	
SEVERITY	Low
WHAT DOES THIS ERROR MEAN?	Header value is modified in the new version
EXAMPLE	Access Control methods header modified
WHY DOES THIS USUALLY HAPPEN?	1. Misconfigured proxy settings 2. Misconfigured security settings in application
POSSIBLE NEGATIVE IMPACTS	Browser security might be compromised depending on the headers modified
SUGGESTED ACTIONS FOR WARRANTED CHANGE	Verify why the reported headers are modified

ERROR TYPE: HEADER REMOVED	
SEVERITY	Low
WHAT DOES THIS ERROR MEAN?	A header is removed in the new version
EXAMPLE	Access Control header removed
WHY DOES THIS USUALLY HAPPEN?	1. Misconfigured proxy settings 2. Misconfigured security settings in application
POSSIBLE NEGATIVE IMPACTS	Browser security might be compromised depending on the headers removed
SUGGESTED ACTIONS FOR WARRANTED CHANGE	Verify why are the reported headers removed