

Explaining the differences

For now, we can import next functions from the teqblazeUtils file:

- isBidRequestValid
- buildRequestsBase
- buildRequests
- buildPlacementProcessingFunction
- getUserSyncs
- interpretResponse

isBidRequestValid

```
modules > JS smarthubBidAdapter.js > ...
154
155 const isBidRequestValid = (bid = {}) => {
156   const { params, bidId, mediaTypes } = bid;
157   let validParams = Boolean(bidId && params && params.seat && params.token);
158
159   if (mediaTypes && mediaTypes[BANNER]) {
160     validParams = validParams && Boolean(mediaTypes[BANNER] && mediaTypes[BANNER].si
161   } else if (mediaTypes && mediaTypes[VIDEO]) {
162     validParams = validParams && Boolean(mediaTypes[VIDEO] && mediaTypes[VIDEO].play
163   } else if (mediaTypes && mediaTypes[NATIVE]) {
164     validParams = validParams && Boolean(mediaTypes[NATIVE]);
165   } else {
166     validParams = false;
167   }
168   return validParams;
169 }
170

libraries > teqblazeUtils > JS bidderUtils.js > buildPlacementProcessingFunction
110
111 export const isBidRequestValid = (keys = ['placementId', 'endpointId'], mode) => (bid = {}) => {
112   const { params, bidId, mediaTypes } = bid;
113   let valid = Boolean(bidId && params && checkIfObjectHasKey(keys, params, mode));
114
115   if (mediaTypes && mediaTypes[BANNER]) {
116     valid = valid && Boolean(mediaTypes[BANNER] && mediaTypes[BANNER].sizes);
117   } else if (mediaTypes && mediaTypes[VIDEO]) {
118     valid = valid && Boolean(mediaTypes[VIDEO] && mediaTypes[VIDEO].playerSize);
119   } else if (mediaTypes && mediaTypes[NATIVE]) {
120     valid = valid && Boolean(mediaTypes[NATIVE]);
121   } else {
122     valid = false;
123   }
124
125   return valid;
126 };
127
```

It is very important for us to check **seat** and **token**
We are also hindered by the `checkIfObjectHasKey` function

functions **buildRequests**, **buildRequestsBase**, **buildPlacementProcessingFunction** have significant differences:

```
modules > JS smarthubBidAdapter.js > spec
171 const buildRequests = (validBidRequests = [], bidderRequest = {}) => {
172   // convert Native ORTB definition to old-style prebid native definition
173   validBidRequests = convertOrtbRequestToProprietaryNative(validBidRequests);
174   const tempObj = {};
175
176   const len = validBidRequests.length;
177   for (let i = 0; i < len; i++) {
178     const bid = validBidRequests[i];
179     const data = _getPlacementReqData(bid);
180     tempObj[data.partnerName] = tempObj[data.partnerName] || [];
181     tempObj[data.partnerName].push(data);
182   }
183
184   return Object.keys(tempObj).map(key => {
185     const request = _buildRequestParams(bidderRequest, tempObj[key]);
186     return {
187       method: 'POST',
188       url: _getUrl(key),
189     };
190   });
191
libraries > teqblazeUtils > JS bidderUtils.js > ...
184
185 export const buildRequests = (adUrl) => (validBidRequests = [], bidderRequest = {}) => {
186   const placementProcessingFunction = buildPlacementProcessingFunction();
187
188   return buildRequestsBase({ adUrl, validBidRequests, bidderRequest, placementProcessingFunction });
189 };
190
191 export const interpretResponse = (serverResponse) => {
192   let response = [];
193   for (let i = 0; i < serverResponse.body.length; i++) {
194     let resItem = serverResponse.body[i];
195     if (isBidResponseValid(resItem)) {
196       const advertiserDomains = resItem.adomain && resItem.adomain.length ? resItem.adomain : [];
197       resItem.meta = { ...resItem.meta, advertiserDomains };
198
199       response.push(resItem);
200     }
201   }
202 }
```

we currently do not use the **getUserSyncs** function

the `interpretResponse` function looks very similar at first glance, but please pay attention to the nested function

```
modules > JS smarthubBidAdapter.js > [e] interpretResponse
193
194 const interpretResponse = (response) => {
195   let responseArr = [];
196   for (let i = 0; i < response.body.length; i++) {
197     let resItem = response.body[i];
198     if (_isBidResponseValid(resItem)) {
199       const advertiserDomains = resItem.adomain && resItem.adomain.length ? resItem.adomain : [];
200       resItem.meta = { ...resItem.meta, advertiserDomains };
201     }
202     responseArr.push(resItem);
203   }
204 }
205 return responseArr;
206 }
207

libraries > teqblazeUtils > JS bidderUtils.js > [e] interpretResponse
190
191 export const interpretResponse = (serverResponse) => {
192   let response = [];
193   for (let i = 0; i < serverResponse.body.length; i++) {
194     let resItem = serverResponse.body[i];
195     if (isBidResponseValid(resItem)) {
196       const advertiserDomains = resItem.adomain && resItem.adomain.length ? resItem.adomain : [];
197       resItem.meta = { ...resItem.meta, advertiserDomains };
198     }
199     response.push(resItem);
200   }
201 }
202
203 return response;
204 };
```

we use additional checks that are not in the proposed replacement

```
modules > JS smarthubBidAdapter.js > [e] _isBidResponseValid
30
31 const _isBidResponseValid = (bid) => {
32   if (!bid.requestId || !bid.cpm || !bid.creativeId || !bid.ttl || !bid.currency || !bid.hasOwnProperty('netRevenue')) {
33     return false;
34   }
35   switch (bid.mediaType) {
36     case BANNER:
37       return Boolean(bid.width && bid.height && bid.ad);
38     case VIDEO:
39       return Boolean(bid.width && bid.height && (bid.vastUrl || bid.vastXml));
40     case NATIVE:
41       return Boolean(bid.native && bid.native.impressionTrackers && bid.native.impressionTrackers.length);
42     default:
43       return false;
44   }
45 }
46

libraries > teqblazeUtils > JS bidderUtils.js > [e] getBidFloor > [e] bidFloor
6
7 const isBidResponseValid = (bid) => {
8   if (!bid.requestId || !bid.cpm || !bid.creativeId || !bid.ttl || !bid.currency || !bid.hasOwnProperty('netRevenue')) {
9     return false;
10  }
11
12  switch (bid.mediaType) {
13    case BANNER:
14      return Boolean(bid.width && bid.height && bid.ad);
15    case VIDEO:
16      return Boolean(bid.vastUrl || bid.vastXml);
17    case NATIVE:
18      return Boolean(bid.native && bid.native.impressionTrackers && bid.native.impressionTrackers.length);
19    default:
20      return false;
21  }
22 };
```